Accurate Restoration of DNA Sequences

by

Gary A. Churchill

BU-1309-M

Received October 1995

Accurate Restoration of DNA Sequences

Gary A. Churchill, Cornell University

1 Introduction

1.1 DNA

DNA is the genetic material in most organisms. It plays a central role in the regulation of cellular function and development and acts as the repository for the hereditary information that is passed from generation to generation. The DNA molecule is a polymer composed of subunits called nucleotides or bases. There are four different nucleotide subunits, denoted by A, C, G and T. The information in a DNA sequence is encoded in the specific ordering of the bases. The familiar double helical structure was first proposed by Watson and Crick (1953). The two strands of DNA are antiparallel in orientation and complementary, such that an A on one strand is always paired with a T on the opposite strand and G is always paired with C. Thus the complete information is contained in the sequence of one strand. The complementary base pairing is the key to DNA replication and other functions and is exploited by many of the technologies used to manipulate DNA molecules, including sequencing reactions. For more information on the structure and functions of the DNA molecule see. for example, the text by Lewin (1992).

The entire DNA content of an organism is called its genome. The human genome is composed of approximately 3 billion base pairs (bp) of DNA that is organized into 24 pairs of chromosomes. A typical chromosome contains a single DNA molecule of 150 million bp. Lengths of DNA sequences are often reported in units of thousands of base pairs (kb) or millions of base pairs (Mb). One of the goals of the Human Genome Initiative is to determine the entire DNA sequence of a typical human being as well as the genomic sequences of a number of experimental organisms.

1.2 Large Scale DNA Sequencing

Methods to determine the sequence of DNA molecules rapidly and at relatively low cost have been available for nearly 20 years (Sanger et al. 1977. Maxam and Gilbert, 1977). Innovations in sequencing technology and increased automation have improved the speed and reliability of these methods while at the same time reducing costs. At present, automated sequencing devices are available that have the potential to produce up to 3 million bases of raw DNA sequence

data per machine per year (Hunkapillar et al. 1991). With further advances expected in the near future, the possibility of determining the entire genomic DNA sequence of a typical human being as well as the genomic sequences of a number of experimental organisms is becoming a reality. Such undertakings will require significant changes in the scale of sequencing projects. The relative costs and quality of sequence data must be carefully considered and it is likely that some form of automated quality control will be implemented as an integral part of these projects. The purpose of this manuscript is to describe in general terms the process of large scale DNA sequencing and to define the potential role of statistical inference and Bayesian methods in a large scale sequencing project.

The stages of a large scale sequencing project are described here. For a more detailed discussion see the review article by Hunkapillar et al. (1991). The first stage of a DNA sequencing project is necessarily the isolation of DNA from the genome of interest and preparation of the DNA for subsequent sequencing steps. This involves (at least) two levels of fragmenting the DNA into manageable sized pieces. Cloning, the first level, is a process by which large DNA segments can be inserted into a host (e.g., a bacterium or a yeast) for maintenance and storage. Subcloning, the second level, involves the production of smaller DNA fragments that are suitable for sequencing reactions. (Problems of assembling these pieces into ordered overlapping sets recur at both levels.) The subcloned DNA fragments are then subjected to sequencing reactions. The resulting reaction products can be separated by size on an electrophoretic gel and the order of the DNA bases determined. Each sequencing reaction can produce only a relatively short (300 to 500 bases) DNA sequence and these sequence fragments must be assembled to reconstruct the original DNA sequence. The finished sequence can then be analyzed to determined its function(s), for example, any protein–encoding genes should be identified and characterized.

1.2.1 Cloning

Large DNA molecules (e.g. entire chromosomes) are difficult to handle experimentally and must be broken into smaller segments that can be maintained and manipulated. DNA segments can be inserted into other DNA molecules, cloning vectors, that can be grown and propagated in a host organism. A variety of cloning vectors are available, each with its own characteristic insert size ranging from 15kb to 1Mb. The foreign DNA insert is called a clone. If the relative overlaps of clones in a collection can be determined, they can be assembled into an ordered overlapping set. This is the first level of the assembly problem and the resulting ordered clone collection forms the basis of a large scale sequencing project. Individual clones can be selected for sequencing and eventually the entire DNA sequence of a genome or large genomic region can be reconstructed.

Once a particular clone has been selected for sequencing, it is necessary to break it into smaller subclones. Typical subclones may be 500 to 2000 bases in size. Several hundred bases of the subclone can usually be determined from a single sequencing reaction. This second level of the assembly problem involves piecing together the fragment sequences and is discussed in detail below.

A number of distinct strategies are available to generate subclones for se-

quencing. Two general classes are the random strategy (shotgun sequencing) and directed strategies. Most sequencing projects begin with a random strategy to rapidly accumulate data in the early stages and then switch to directed approaches to close any remaining gaps. In a shotgun sequencing project (Messing et al. 1981), many copies of a large DNA molecule (clone) are randomly broken into a collection of fragments. Thus the sequences obtained have random startpoints along the clone and random orientations. Assembly of the fragments is a problem and gaps may remain in regions where, by chance, no subclones were generated. In a purely directed strategy, the sequence information in a previously sequenced subclone is used to generate the next subclone such that the two overlap by 25% to 50%. Thus the relative location of each subclone is known in advance and one can "walk" from one end of a clone to the other. Directed strategies typically require more human effort than random strategies. An example of a partially directed strategy is to use large subclones that can be sequenced inward from both ends. Thus partial information about their relative placement is available to aid the assembly process (Lawrence et al., 1994).

1.2.2 Sequencing

The most widely used sequencing technologies are based on the enzymatic method of Sanger et al. (1977). Four separate reactions are carried out; the products of each reaction are partial copies of the fragment being sequenced ending at a base of known type. Thus four nested series of molecules are generated. One series contains all partial copies of the DNA molecule ending at an A in the template sequence. The other series contain all partial copies ending at C, G and T. The reaction products are separated by size using gel electrophoresis, a process in which DNA molecules move through a gel under the influence of an electric field. Smaller molecules move faster through the gel. Molecules that differ in size by one base can be resolved over a size range that will typically yield 300 to 500 bases of sequence information. (A picture of such a gel is shown in figure 1.) Advances in electrophoresis technology may soon yield runs of up to 1000 bases. The linear order of bases in the sequence can be read off as the reaction products are detected moving through the gel.

Automated DNA sequencers are capable of running several reactions in parallel. Reaction products are labeled with a fluorescent dye and are detected as they pass a scanning laser. The resulting "traces" can be fed directly into a computer and interpreted to yield a sequence of "base calls". The base calling algorithms present some interesting statistical problems that will not be addressed here. For example, Bowling et al. (1991), have applied a neural network approach to interpreting traces. Their approach combines the peak heights and the phase information from the traces to improve the accuracy of base calls. Another interesting and open problem would be to interpret the traces to yield probabilistic base calls. In this way some measure of accuracy would be available in the raw data. Some recent work by Lawrence and Solovyev (1994) has addressed this problem. See section 5 for further discussion. In the present work, we assume the base calls are given as A, G, C, T or in the case of ambiguity as N.

Figure 1. Photograph of a
sequencing gel with four lanes
reactions shown. Each set of four lanes
shows the ordering (top to bottom) of A, C, T and G
bases in a DNA sequence.

1.2.3 Assembly

We will address the problems of assembling a set of DNA sequence fragments and determining the sequence of a clone below. An interleaving of the fragments must be determined by identifying overlaps among their sequences. The fragments are then assembled and aligned to form a column–by–column correspondence and any ambiguities in the overlapping portions must be resolved. Finally a consensus sequence, an estimate of the clone sequence, is inferred from the assembled fragments. The assembly problem has been addressed and software for assembly has been produced by Staden (1980), Kececioglu and Meyers (1990) and Huang (1992). Despite these efforts, assembly remains a major bottleneck in most large scale sequencing projects.

The problem of assembling DNA fragments is similar in many respects to the (higher–order) problem of assembling a collection of clones into an ordered overlapping set. This problem has been studied, for example, by Michiels et al. (1987), Lander and Waterman (1988), Branscomb et al. (1990), Balding and Torney (1991), Fu et al. (1992) and Alizadeh et al. (1992). Many of these results are directly applicable to the fragment assembly problem.

Two important quantities that arise in both assembly problems are closure and the redundancy of coverage. Closure is the proportion of bases in the clone that have been sequenced at least once. Coverage is a base–by–base measure of the number of times a base has appeared in a sequenced fragment. Average coverage is often reported as a measure of progress of a sequencing project. Coverage will vary for statistical as well as biological reasons.

1.2.4 Analysis

The final stage of sequencing project is the interpretation of the finished sequence data to determine its function(s). This is another aspect of DNA sequencing in which many interesting statistical problems arise. Sequence analysis problems will not be addressed here. A Bayesian approach to detecting coding sequences in finished sequences is described by States and Botstein (1992). A number of other authors have considered the analysis of DNA sequences that are likely to contain errors (see Borodovsky and McIninch, 1992; Clark and Whittam, 1993).

1.3 Examples

In this section we review some recent large scale sequencing efforts with an emphasis on sequencing strategies and quality control. Raw sequencing data are available from some of these projects as noted below.

The complete sequence of yeast chromosome III (Oliver et al. 1992) is presently the largest known contiguous DNA sequence. This sequence was determined by a consortium of laboratories using various techniques. The final 315 kilobases (kb) sequence was obtained from a total of 385kb of sequence provided by different laboratories. Thus about 20% overlap is present in the sequence and provides an opportunity to examine the accuracy of sequence data. When overlapping regions derived from the same strain were compared the rate of disagreements

was about 0.0004 per base. Comparison of sequences obtained from different strains reveals disagreements at a rate of about .0062 per base, much of which may be attributable to naturally occurring variation in the DNA. Other checks on the quality of this sequence suggest that it is highly accurate, to at least an order of magnitude of 0.001 errors per base.

Edwards et al. (1990) reported the sequence of a 57kb region of human DNA containing a gene for the enzyme HPRT. DNA was obtained from six clone sequences. An initial stage of random sequencing was carried out to achieve 96% closure of the region and was followed by directed strategies to obtain full closure. The average redundancy in the finished sequence is 4 times. No estimates of error rates are provided, but Edwards et al. (1990) do point out that each redundant base represents independent cloning, sequencing and reading events thus reducing the potential for error and aiding resolution of compressions and other artifacts. The coverage is summarized in figure 2. The six large groupings in figure 2 correspond to the six clone sequences with the redundancy of these clones indicated by their overlap. The small arrows within clones represent the individual sequence fragments derived from subclones; their aggregation on the plot indicates overlapping and redundancy in sequencing the clones. Ambiguity rates in the assembled fragments from this project were studied by Huang (1992).

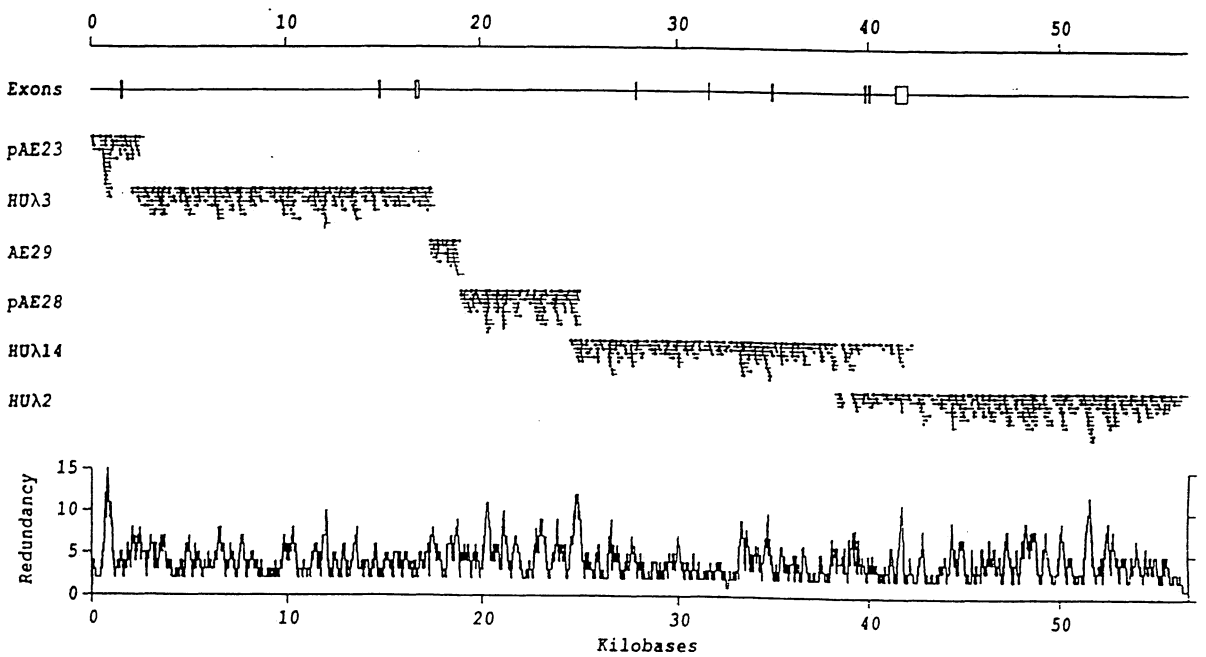

Figure 2: Coverage of the Human HPRT sequence (reproduced with permission from Edwards et al. 1991). The position size and orientation of each sequencing gel is shown for 6 clone sequences.

Chen et al. (1991) reported a 20kb sequence of human DNA containing a gene encoding the enzyme G6PD. DNA was isolated from three clone sequences. A random strategy was used to obtain 86.2% closure for the largest of the three clones (11768bp) followed by directed strategies to obtain full closure. The average coverage was 3.5 times. The two smaller fragments (3483bp and 4863bp) were sequenced by purely directed strategies with an average coverage of 2.5 times. Chen et al. (1991) favored the directed strategy for small clones because of the reduced redundancy, better organization of the raw data and better accuracy. They "estimate from the quality of the sequencing gels and the agreement of repeated threefold determinations on overlapping cloned fragments that the sequence of the 20114bp is determined with 99.9% precision." These data were analyzed by Churchill and Waterman (1992) who reached a similar conclusion.

Daniels et al. (1992) reported the sequence of a 91.4kb region of the *E. coli* genome. The region was contained in 9 clones and represents about 2% of the entire *E. coli* genome. An initial random stage was carried out to an average coverage of 6 times and was followed by directed closure. To ensure accuracy of the finished sequence, a minimum coverage of 4 times was obtained for 95% of the region and 90% was sequenced at least once in each orientation. The average coverage in the finished sequence was 9.2 times. Daniels et al. (1992) reported that, after the initial (automated) assembly, ambiguities occurred at a rate of 1 per 100bp. Human editing of the assembly reduced this to 1 per 200bp and data proofreading combined with genetic analysis brought this down to a final rate of 1 per 600bp. They report an "internal estimate of 1 error per 500 residues" in their finished sequence. We note that the human proofreading was a very time–consuming aspect of this project.

Sulston et al. (1992) report results from sequencing three clones containing DNA from the genome of *C. elegans*. They explored a number of different sequencing strategies and methods that included different proportions of random versus directed sequencing strategies. They reported on several types of sequencing errors and broke down rates by position within fragments. The problem of position–dependent errors is discussed in Section 4.

Seto et al. (1992) have made publicly available the raw sequencing data from a human DNA region encoding a T–cell receptor protein. The data consist of 1023 raw and 820 partially refined fragment sequences as well as a 34476bp derived consensus sequence. They propose this to be a test data set for the (fair) comparison of different assembly algorithms. This data has been assembled and analyzed by Huang (1992). The process of refining raw fragment sequences opens some interesting statistical questions not addressed here. For example, it is not clear what, if any, advantage is gained by trimming the raw fragment sequences. The intuitive idea is to eliminate unreliable base calls from the raw data. Since these occur most frequently near the beginning and end of a fragments sequence, rules have been developed to trim ends that contain an excess of ambiguous base characters (N's).

These examples of large–scale sequencing projects demonstrate the need for objective and statistically sound estimates of sequence quality. They indicate that current technologies are able to produce sequences of 50 to 500kb in length

with error rates on the order of 0.001 per nucleotide. Most reports indicate that a major bottleneck in the sequencing process is presented by the need to store, assemble and analyze the raw data. At present, assembly is a time–consuming step that requires a great deal of human intervention and rechecking of the raw data. The computational and statistical problems involved are still largely unsolved.

1.4 Overview

Current practice in sequence restoration is to use a mixture of *ad hoc* algorithms and human editing to produce an assembly of the fragment sequences. A consensus of the fragments is constructed with little or no consideration given to error rates or accuracy and is reported as the finished sequence. To address the accuracy problem, Churchill and Waterman (1992) developed an EM algorithm approach that assumes the sequence fragments are assembled correctly. The clone sequence is restored conditional on a maximum likelihood estimator (MLE) of the error rates. In the present work we wish to avoid both of these assumptions (correct assembly of fragments and known error rates). Thus our goal is to estimate the true DNA sequence from its posterior distribution, marginal over the assembly and the error rate parameters. Monte Carlo methods appear to provide a practical approach to this problem. Several variations are possible. We describe one that seems promising.

In Section 2, we look at the problem of assembling DNA sequence fragments into an interleaving from which the underlying clone sequence can be deduced. A detailed solution to the problem will not be provided. Instead we will identify where problems arise and suggest some general and rather open–ended approaches. In Section 3, the problem of estimating and assessing the (post-data) accuracy – taking as given the method of assembling fragments – of a finished DNA sequence is addressed. Here we do provide a detailed solution for a simplified version of this problem. In Section 4 we consider some of the experimental realities and suggest directions in which the simple model might be generalized. Some prospects for future work on the DNA accuracy problem are presented in Section 5.

2 Fragment Assembly

In this section, we provide a brief overview of the fragment assembly problem and emphasize the potential role of Bayesian methods in its solution. Section 3 does not build directly on this material and may be read independently.

Assembly of a set of DNA sequence fragments is accomplished by determining the overlaps among the fragments in the set and using this information to position them relative to one another. The potential role of Bayesian methods and in particular Monte Carlo solutions are significant and underdeveloped at present. The subproblem of determining pairwise overlaps among fragments has a natural prior distribution and in some simple cases it is possible to derive closed form expressions for the posterior (Churchill, unpublished). A posterior distri-

bution for the full assembly however appears to be intractable. Approximate solutions have been proposed (Alizadeh et al. 1992) but further developments are needed before a fully satisfactory and practical solution to this problem can be achieved.

2.1 The Lander–Waterman Model

Lander and Waterman (1988) proposed a stochastic model for the process of assembling a clone collection. Their model can also be applied to the DNA fragment assembly problem when a random sequencing strategy has been used. Although the model is very simple it seems to provide robust predictions of the behavior of actual sequencing projects (e.g., Chen et al. 1991, Edwards et al. 1990).

Consider a set of fragments generated by a random sequencing strategy. Let

$$\begin{aligned} G &= \text{length of the clone in base pairs,} \\ L &= \text{length of a fragment in base pairs and} \\ N &= \text{the number of fragments.} \end{aligned}$$

We will assume that all these quantities are known and that L is constant for all fragments. The Lander–Waterman model specifies that the *a priori* placements of the left–hand endpoints of the fragments are independent and uniformly distributed in the interval $[0, G - L]$.

Under these assumptions, the redundancy of coverage (i.e. the number of fragments into which a base in the clone sequence is copied) will behave as a queuing process across the bases of the clone. (See Taylor and Karlin, 1984, p. 353, for a description of the $M/G/\infty$ queuing model.) In practice, coverage will vary for biological as well as statistical reasons (e.g. Edwards et al. 1990). Lander and Waterman (1988) describe the behavior of this process in terms of the expected number of "islands", which are overlapping sets of fragments (also known as "contigs"), and "oceans", which are the gaps between islands. A critical factor in the assembly process is our ability to detect those overlaps that actually exist between fragments (and to avoid false–positive overlaps). If T is the actual overlap (in base pairs) between two fragments, Lander and Waterman assume that when $T/L > t$ the overlap will be detected and otherwise not. The expected number of apparent islands after N fragments have been assembled is $N \exp\{-c(1-t)\}$, where $c = NL/G$ is the average coverage. In a typical sequencing project the value of t may be 0.10 to 0.20 and the coverage c may range from 2 to 10.

Closure is defined to be the proportion of bases in the clone sequence that are copied in at least one fragment. A simple geometric argument (Clarke and Carbon, 1976) yields the expected closure.

$$\begin{aligned} E(\text{closure}) &= 1 - (1 - L\ G)^N && (1) \\ &\approx 1 - e^{-NL/G}. && (2) \end{aligned}$$

Both the expected closure and expected numbers of islands are pre–data measures that indicate how near to completion a sequencing project is. An interesting open problem would be to develop post-data measures that could be used

to indicate progress of an ongoing project more accurately in light of the data accumulated thus far.

In the early stages of a random sequencing project, new sequence information is accumulated rapidly but as the project progresses the sequences become increasingly redundant. Edwards et al. (1990) recommend carrying the random stage to 95% closure and then switching to directed strategies. At the end of the random stage of the G6PD sequencing project (Chen et al. 1991), $G = 11768$, $L = 265$ and $N = 145$. Thus the average coverage is 3.27 and the expected closure is 0.963. For the HPRT project (Edwards et al. 1990) $G = 56736$, $L = 265$ and $N = 695$. Thus the average coverage is 3.25 and the expected closure is 0.961. It could be of significant practical interest to frame the sequencing strategy problem in a decision theoretic context. Sulston et al. (1992) investigated a number of switching rules (random strategy to directed strategy) based on practical considerations of manpower and equipment usage.

2.2 Pairwise Comparison Methods

Fingerprinting is a term that refers to any characterization of a clone or DNA sequence fragment. The forms of fingerprinting data for the clone assembly problem are highly varied and likely to change as new experimental methods of characterizing clones are developed. For the fragment assembly problem, the sequence itself is a highly informative fingerprint. In this context, the fingerprint is used to determine probable overlaps between pairs of fragments.

Pairwise comparisons among the fragments play a central role in the assembly process. In principle, probabilities of higher order relationships, e.g. among triplets of fragments, could be computed. It may be worthwhile to investigate how much additional information is gained by computing such probabilities as the computation is likely to be expensive. In this section we discuss the prior probability of pairwise overlap, and the posterior probability of overlap given the fingerprints of two clones or sequence fragments.

Prior probability of overlap. Consider two fragments selected at random from a collection of N fragments and let $T \in [0, L]$ denote the actual overlap (in base pairs) between them. A geometric argument built upon uniformity assumptions yields the prior probability distribution of T,

$$\Pr(T \le t) = \begin{cases} \left(\frac{G-2L}{G-L}\right)^2 & t = 0 \\ \left(\frac{G-2L+t}{G-L}\right)^2 & t > 0. \end{cases}$$

By ignoring "edge effects" (i.e. assume $G \gg L$), we obtain a simple approximate prior

$$\Pr(T \le t) = \begin{cases} 1 - \frac{2L}{G} & t = 0 \\ 1 - \frac{2}{G}(L - t) & t > 0. \end{cases}$$

The (approximate) prior density function has a point mass of $1 - 2L/G$ at zero and constant density $2/G$ on $0 < t \le L$. The prior probability of any overlap between two randomly chosen fragments is $\Pr(T > 0) = 2L/G$.

Likelihood and Posterior To illustrate the problem of determining pairwise overlap probabilities, we consider a simple case. Consider a word w_i in the DNA alphabet, for example $ACCTGT$. For each fragment $\mathrm{f}_j, j = 1, \ldots, m$ in the set, we can observe the binary outcome

$$X_{ij} = \begin{cases} 1 & w_i \text{ is present in fragment } \mathrm{f}_j \\ 0 & \text{otherwise} \end{cases}$$

As a first approximation, we will assume that the locations of the first letter of each occurrence of the word w_i are distributed uniformly throughout the clone sequence with known rate λ_i and that there are no errors in the process of copying fragments from the clone.

We wish to compare two fragments j_1 and j_2. Let $A_{ij_1j_2} = X_{ij_1} + X_{ij_2}$ and suppress the subscripts j_1 and j_2. Then

$$A_i = \begin{cases} 0 & \text{if neither fragment contains } w_i, \\ 1 & \text{if exactly one fragment contains } w_i, \text{ and} \\ 2 & \text{if both fragments contain } w_i. \end{cases}$$

The posterior probability distribution of T given A_i is then $\Pr(T \mid A_i) \propto \Pr(A_i \mid T)\Pr(T)$ where, by a geometric argument, the likelihood terms are

$$\begin{aligned} \Pr(A_i = 0 \mid T = t) &= e^{-\lambda_i(2L-t)} \\ \Pr(A_i = 1 \mid T = t) &= 2e^{-\lambda_i L}(1 - e^{-\lambda_i(L-t)}) \\ \Pr(A_i = 2 \mid T = t) &= 1 - e^{-\lambda_i t} + e^{-\lambda_i L}(1 - e^{-\lambda_i(L-t)})^2 \end{aligned}$$

Closed form expressions for the posterior can be derived in this simple case. The procedure can be repeated using a set of words and the overlap probabilities can be updated using Bayes rule (under the assumption that occurrences of words are independent).

A more challenging problem is to use the information in the entire fragment sequence to determine the posterior probability of overlap. We are currently working to adapt the methods of Thorne et al. (1991) to compute overlap probabilities by summing over all possible alignments between a fragment pair. Huang (1992) describes a screening method that can quickly eliminate pairs of fragments that are very unlikely to overlap. He uses a standard alignment method that yields a score that could be interpreted (up to a constant term) as the log-likelihood of the best pairwise alignment. The choice of an informative and easy–to–compute fingerprinting method and the calculation of posterior probabilities remain open problems.

2.3 Full posterior of an assembly

An assembly can be represented as an interleaving of fragments. If there are no errors in the raw fragments, it will be sufficient to specify the left–hand endpoint of each fragment in some global coordinate system. Let $\mathbf{x} = \{x_1, \ldots, x_n\}$ denote the left endpoints of n fragments. Alizadeh et al. (1993) look at the problem of computing the posterior probability of an interleaving I given fingerprint data D in the context of clone assembly. They show that $\Pr(I \mid D)$

is proportional to $\int_{K(I)} \Pr(D \mid \mathrm{x}) \partial \mathrm{x}$ where $K(I)$ is a polyhedral set in R^n. An exact solution would appear to be intractable. However it is likely that reasonable approximate solutions can be developed. Alizadeh et al. (1993) note that the assembly problem is NP–complete and can be formulated as a traveling salesman problem. They discuss an approach based on the stochastic optimization of a function approximating $-\log \Pr(I \mid D)$. Monte Carlo methods are used to produce multiple near–optimal solutions. Ideally, weights that approximate posterior probabilities should be assigned to these alternative solutions. Again, the problem of finding computationally feasible solutions to the posterior probability of an assembly remains as a challenging open problem.

3 Restoration of the Clone Sequence

In section 2 we discussed several open problems in assembling fragments. In this section we will assume that an initial assembly of the fragments has been generated and address the problems of restoring the clone sequence and assessing its accuracy. We begin section 3.1 by introducing a simple stochastic model of fragment generation. Concepts of sequence alignment and fragment assembly are introduced and used to define the likelihood of a fragment set. A sampling algorithm is described that can be used to obtain an approximate posterior distribution for the clone sequence. The remaining sections (3.2–3.4) develop the details for each of the steps in the resampling algorithm. These are algorithms that sample the clone sequence (section 3.2), the error rate parameters (section 3.3) and the fragment assembly (section 3.4) from conditional posterior distributions given the fragment sequences and the other two quantities.

The clone is a DNA molecule with a unique but unknown sequence denoted by $\mathrm{s} = (s_1, \ldots, s_{n_s})$. The length of the clone sequence n_s is unknown, but may typically be on the order of 15 to 40 thousand bases. The individual bases of the clone sequence are elements of the alphabet $\mathcal{A} = \{A, C, G, T\}$. Thus s is an element of the set $\mathcal{S} = \bigcup_{k=1}^{\infty} \mathcal{A}^k$, where $\mathcal{A}^k$ are the sets of k–tuples on the alphabet $\mathcal{A}$. The set $\mathcal{S}$ will be referred to as sequence space and is the space on which we will define the posterior distribution.

The observed data are sequence fragments obtained from subclones of the clone sequence. The set of fragment sequences will be denoted by $\mathrm{F} = \{\mathrm{f}_1, \ldots, \mathrm{f}_m\}$, where $\mathrm{f}_j = \left(f_{j1}, \ldots, f_{jn_j}\right)$ and n_j is known (typically 250 to 1000 bases). Each fragment sequence is generated by an automated sequencing device as a series of base calls drawn from the alphabet $\mathcal{B} = \{A, C, G, N, T\}$ which includes the ambiguous character N in addition to the four standard DNA bases.

3.1 The Copying Process

3.1.1 A Hidden Markov Model

In this section we define a hidden Markov model (HMM) that describes the process of generating a single fragment $\mathrm{f} = \left(f_1 \ldots f_{n_f}\right)$ by copying a subsequence of bases in s. (Because we are considering only one fragment, the subscript j

will be suppressed.) We make certain simplifying assumptions here and discuss generalizations of the model in section 4. For now, we assume that all fragments are generated in the same orientation from the clone sequence (i.e., from left to right). The problem of reversed complement copies will be addressed in section 4.1. We also assume that the parameters of the copying process are constant across all fragments, all bases within a fragment and all bases within the clone. Generalizations of this assumption will be addressed in section 4.2. The model is summarized in figure 3 and is described further here.

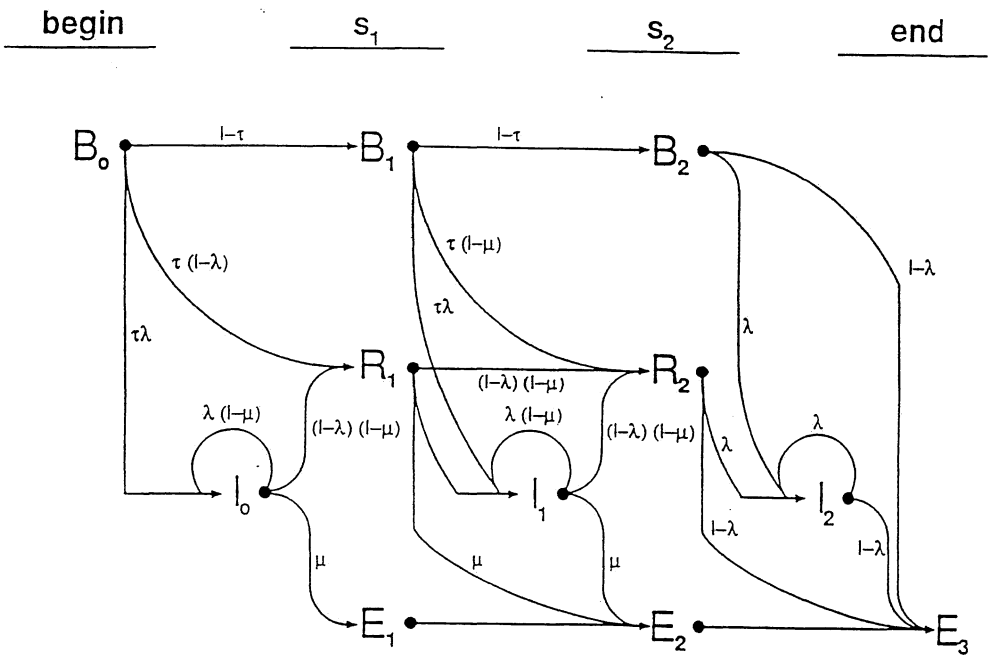

Figure 3: A hidden Markov model representation of the copying process is shown for a clone sequence of length $n_s = 2$. The process begins in state B_0 and terminates in state E_3. Possible state transitions are shown by arrows connecting the states B_i, R_i, I_i and E_i. Transition probabilities are indicated along each arrow. The observed fragment sequence is generated as a series of outputs by the states R_i and E_i.

Each base s_i in the clone is associated with 4 hidden states in the HMM, B_i, R_i, I_i and E_i. In addition there are two states, B_0 and I_0, associated with the start of the clone and a state E_{n_s+1} associated with the end of the clone. The copying process starts in state B_0. All of the states are non–recurrent except E_{n_s+1}, which is absorbing. The notation is chosen to suggest the role of each state in the copying process. Any uncopied clone bases starting from the leftmost (s_1) to the first base that is copied are associated with B–states ("begin"). Bases in the clone sequence that are copied, replaced with another base or deleted in the fragment are associated with R–states ("replace"). Bases that may be inserted into the fragment during the copying process are generated by I–states ("insert"). Any uncopied clone bases beyond the last base copied to the rightmost base (s_{n_s}) are associated with E–states ("end").

Transition probabilities between states are indicated in figure 3. The full state transition matrix has dimension $(4n_s+3)^2$ but its block diagonal structure can be summarized by the following partial transition matrices. The initial block has the form

$$\begin{array}{c} \\ B_0 \\ I_0 \end{array} \begin{array}{c} \begin{array}{cccc} B_1 & R_1 & I_0 & E_1 \end{array} \\ \begin{bmatrix} 1-\tau & \tau(1-\lambda) & \tau\lambda & 0 \\ 0 & (1-\lambda)(1-\mu) & \lambda(1-\mu) & \mu \end{bmatrix} \end{array}$$

The main body of the transition matrix can be written as

$$\begin{array}{c} \\ B_i \\ R_i \\ I_i \\ E_i \end{array} \begin{array}{c} \begin{array}{cccc} B_{i+1} & R_{i+1} & I_i & E_{i+1} \end{array} \\ \begin{bmatrix} 1-\tau & \tau(1-\lambda) & \tau\lambda & 0 \\ 0 & (1-\lambda)(1-\mu) & \lambda(1-\mu) & \mu \\ 0 & (1-\lambda)(1-\mu) & \lambda(1-\mu) & \mu \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

and the terminal block is

$$\begin{array}{c} \\ B_n \\ R_n \\ I_n \\ E_n \end{array} \begin{array}{c} \begin{array}{cc} I_n & E_{n+1} \end{array} \\ \begin{bmatrix} \lambda & 1-\lambda \\ \lambda & 1-\lambda \\ \lambda & 1-\lambda \\ 0 & 1 \end{bmatrix} \end{array} .$$

We note that the rows corresponding to transitions out of R–states and I–states are identical. This constraint on the structure of the model simplifies the restoration problem addressed in section 3.2.

The observed bases in the fragment sequence are generated as outputs by the states of the hidden Markov chain. The states B_i indicate that the copying of the fragment has not yet begun and thus no output is generated. In the state R_i, the clone base s_i is copied. The output of an R–state is generated according to the distribution $\pi_R(b|s_i)$ where $b \in \mathcal{B} \bigcup \{-\}$. The character $-$ is used to indicate a null output and is not directly observable. The event of a null output by state R_i corresponds to the deletion of base s_i in the copied fragment. The states I_i generate insertions into the fragment sequence according to the distribution $\pi_I(b)$ where $b \in \mathcal{B}$. Bases output by state I_i are spurious additional

bases inserted to the right of any output from the state R_i and, by convention, correspond to the base s_i in the clone. Finally the states E_i indicate that copying of the fragment has terminated and no output is generated.

The 32 parameters governing the hidden Markov chain will be denoted by $\theta = \{\tau, \lambda, \mu, \pi_R(\cdot|\cdot), \pi_I(\cdot)\}$ below. Where τ, λ and μ are transition probabilities between states of the HMM, π_R is a 4×6 row multinomial matrix defining the conditional output distribution of a R state given s_i and π_I is a multinomial vector of 5 probabilities defining the output distribution of an I state. Due to the usual multinomial constraints on π_R and π_I, there are 27 degrees of freedom in the model. We note that in other applications of HMMs (e.g., Krogh et al. 1993), the parameter values are not constrained to remain constant from state to state and the resulting HMMs are highly parameterized.

3.1.2 Alignments and Assembly

An *alignment* between a fragment sequence $\mathbf{f}$ and the clone sequence $\mathbf{s}$ is an hypothesis that establishes a correspondence between the individual bases in the two sequences. It can be represented as a directed graph associated with the HMM in section 3.1.1, whose vertices form a grid with $(n_f + 1)$ rows and $(n_s + 1)$ columns (figure 4), where n_f is the length of $\mathbf{f}$ and n_s is the length of s. Let $v(i, j)$ denote the vertex at column i and row j of the graph for $i = 0, \ldots, n_s$ and $j = 0, \ldots, n_f$. The clone sequence is shown across the northern edge of the grid such that base s_i falls between the columns $i - 1$ and i. The fragment sequence is shown down the western edge of the grid so that base f_j falls between rows $j - 1$ and j. An alignment is shown as a path, a connected sequence of arcs, that traverses the matrix from a vertex on its northern edge to a vertex on its southern edge by a series of southern ($\downarrow$), southeastern ($\searrow$) and eastern ($\rightarrow$) moves. A southern arc connecting $v(i, j - 1)$ to $v(i, j)$ indicates that f_j was generated as an insertion by the state I_i. A southeastern arc connecting $v(i - 1, j - 1)$ to $v(i, j)$ indicates that f_j is the non–null output of state R_i, i.e. f_j is copied, perhaps with error, from s_i. An eastern arc connecting $v(i - 1, j)$ to $v(i, j)$ indicates that the output of state R_i was null, i.e. that base s_i was deleted from the fragment.

To indicate the point at which the copying of a fragment begins, we define a set of special vertices $\{v(i, -1) : i = -1, \ldots, n_s - 1\}$ that lie above the northern edge of the path graph and connecting arcs as shown in figure 4. A southeastern arc connecting $v(i - 1, -1)$ to $v(i, 0)$, $i = 0, \ldots, n - 1$ indicates that the one of the transitions $B_i \rightarrow R_{i+1}$ or $B_i \rightarrow I_i$ has occurred and the first base copied from the clone sequence is s_{i+1}. An eastern arc indicates that copying has not yet started. Southern arcs are not allowed here. Similarly, we define a special set of vertices $\{v(i, n_f + 1) : i = 1, \ldots, n_s + 1\}$ that lie below the southern edge of the path graph and connecting arcs to indicate where the copying of a fragment ends. A southeastern arc connecting $v(i, n_f + 1)$ to $v(i + 1, n_f + 1)$, $i = 1, \ldots, n$ indicates that the one of the transitions $R_i \rightarrow E_{i+1}$ or $I_i \rightarrow E_{i+1}$ has occurred and thus that s_i is the last base of the clone to be copied.

With some exceptions to be noted in a moment, the entire alignment path

can be summarized as a sequence of arcs denoted $\vec{\alpha} = \alpha_1, \ldots, \alpha_n$, where

$$\alpha_i = \begin{cases} 0 & \rightarrow & \text{delete} \\ 1 & \searrow & \text{copy} \\ 2 & \downarrow & \text{insert}, \end{cases} \tag{3}$$

and n is the length of the alignment, $\max(n_s, n_f) + 2 \leq n \leq n_s + n_f + 2$. Arcs at the beginning and end of $\vec{\alpha}$ are interpreted differently and handled specially below. The first occurrence of a "1" at index i in the sequence indicates the transition from state B_{i-1} to one of the states I_{i-1} or R_i. All zeros to the left of this point indicate transitions between B–states. The last occurrence of a "1" in the sequence at index j indicates a transition from one of the states R_j or I_j to the state E_{j+1}. All zeros to the right indicate transitions between E–states. Thus $\vec{\alpha}$ defines the sequence of states by the HMM as it generated the observed fragment sequence $\mathbf{f}$.

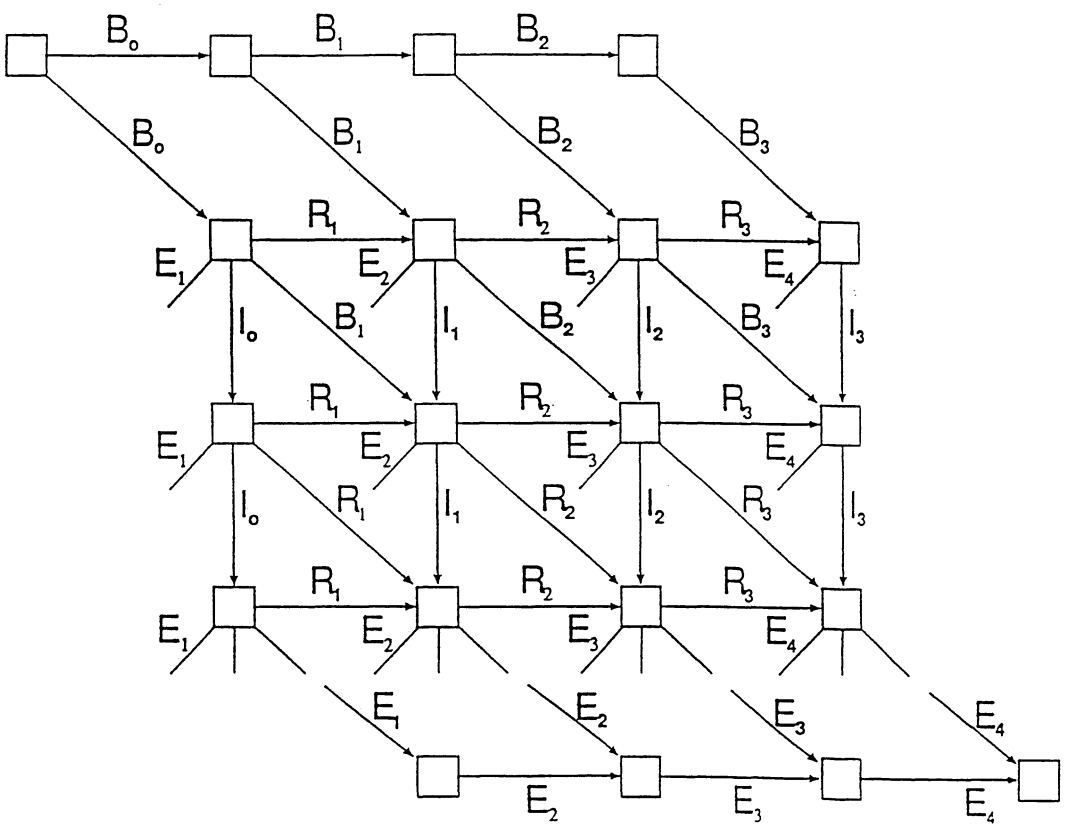

Figure 4: The path graph representation of an alignment is shown. Each arc in the graph corresponds to a unique state in the HMM as indicated. Nodes in the path graph represent tranisition between states.

An *assembly*, denoted $\mathbf{A}$, of the fragment set $\mathbf{F}$ establishes a correspondence among the bases in different fragment sequences. (As we are now discussing the set of fragments, the subscript $j = 1, \ldots, m$ will be used again to index individual fragments and their corresponding alignments.) It can be derived from the set of alignments $\{\vec{\alpha}_1, \ldots, \vec{\alpha}_m\}$. Together the assembly and the fragment set determine the *assembled fragment set*, denoted $\mathbf{X} = \{\mathbf{F}, \mathbf{A}\}$. An example is shown in figure 5. The assembled fragment set is a matrix with elements x_{ij}, $i = 1, \ldots, n_A$ and $j = 1, \ldots, m$ drawn from the alphabet $\mathcal{B} \bigcup \{-, \phi\}$. Note that i denotes columns and j denotes rows. Each row of the matrix contains the complete sequence of a fragment $\mathbf{f}_j$, $j = 1, \ldots, m$ along with the two types of null characters. The null character $-$ is called a gap and the null character ϕ is called an offset and will usually be written as a blank character " ". Gap characters may be inserted inside the fragment sequence or immediately adjacent to either of its ends. Offset characters may be inserted beyond the ends of a fragment sequence. The total number of bases, gaps and offsets in each row of $\mathbf{X}$ is n_A, the width of the assembly.

The placement of gap and offset characters defines a column–by–column correspondence among the bases in different fragment sequences. The columns of $\mathbf{X}$ will be denoted by $\mathbf{x}_i$, $i = 1, \ldots, n_A$. All non–null characters in a column are generated by the same underlying state of the HMM. If the state corresponding to column $\mathbf{x}_i$ is an R–state, the gap characters correspond to deletion events. In the case where all outputs of an R–state are null, there will be no column in the assembly corresponding to that R–state and no indication that a state was missed. This is one way in which an error can arise in a restored clone sequence. If the state corresponding to column $\mathbf{x}_i$ is an I–state, a gap character in row j is a place holder to indicate that the state I_i was not visited in the copying of fragment j but that state I_i was visited in the copying of at least one other fragment. In the event of multiple insertion events, two or more adjacent columns in the matrix $\mathbf{X}$ may correspond to the same I–state. Offset characters to the left of a fragment sequence correspond to B–states and those to the right correspond to E–states. In summary, the assembly $\mathbf{A}$ specifies the locations of offset and gap characters needed to construct the assembled fragments matrix $\mathbf{X}$ from the fragment set $\mathbf{F}$. The offset characters define the beginning and end of the subsequence of $\mathbf{s}$ that was copied in fragment $\mathbf{f}_j$. Gap characters are needed to establish a correspondence among the bases in the fragments when insertions and/or deletions occurred in the copying process.

Because the clone sequence and its correspondence with the columns of $\mathbf{X}$ are not given as part of the assembly, it is not possible to identify which gap characters in the assembled fragments set are deletions and which are place holders for insertions that occurred in other fragments. The alignment set $\{\vec{\alpha}_1, \ldots, \vec{\alpha}_m\}$ contains additional information that establishes a correspondence between the columns of $\mathbf{X}$ and the bases of $\mathbf{s}$. The correspondence information can be summarized as a sequence $\Gamma = \gamma_1, \ldots, \gamma_{n_A}$ where $\gamma_i \in \{0, 1, 2, \ldots\}$ is the number of bases in the clone sequence associated with column i in the assembly. Correspondence between the columns of the assembled fragments matrix and the

clone sequence is established by the following algorithm.

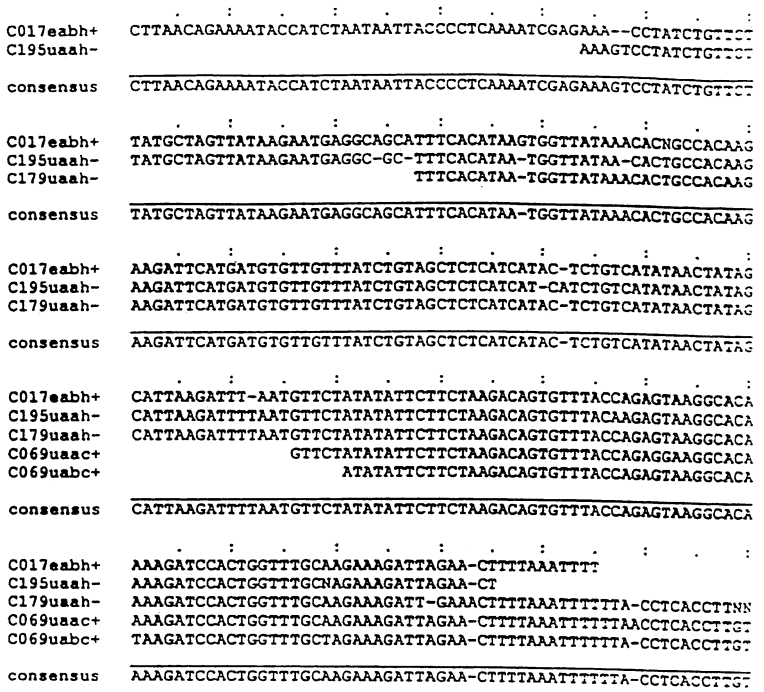

Figure 5: A portion of an assembled fragment set as generated by the computer program CAP (Huang, 1992). Fragment identifiers are shown at the left and the orientation of each fragment is indicated by + (direct) or − (reversed) followed by the fragment sequences. The assembly is broken into blocks of 60 columns in width for display purposes. A majority rule consensus sequence is shown below each block.

Algorithm Let $j = 1$. For $k = 1$ to n_A:

1. If $\gamma_k = 0$, the column $\mathbf{x}_k$ is the output of state I_j. The column corresponds to a gap in the clone sequence. Increment $k = k + 1$.

2. If $\gamma_k = 1$, the column $\mathbf{x}_k$ corresponds to base s_j in the clone sequence. This occurs when at least one output of the state R_j is non–null. All non–offset characters in column $\mathbf{x}_k$ are outputs generated by state R_j. Increment $j = j + 1$ and $k = k + 1$.

3. If $\gamma_k = 2$ or more, the column $\mathbf{x}_k$ corresponds to s_j but there is no column in $\mathbf{X}$ corresponding to the bases $s_{j+1}, \ldots, s_{j+\gamma_k-1}$. This event occurs when all outputs of states $R_{j+1}, \ldots, R_{j+\gamma_k-1}$ are null. Increment $j = j + \gamma_k$, $k = k + 1$.

We note that Γ is an alignment (in the sense defined above) between the clone sequence $\mathbf{s}$ and the columns of the assembled fragments matrix $\mathbf{X}$. The pair $\{\mathbf{A}, \Gamma\}$ is equivalent to the set of alignments $\{\vec{\alpha}_1, \ldots, \vec{\alpha}_m\}$ in the sense that there is no loss of information. Either data structure is completely determined given the other. Thus, we have partitioned the information in the alignment set into an assembly $\mathbf{A}$ and a correspondence vector Γ. Also note that $\sum_{i=1}^{n_A} \gamma_i = n_s$.

The problem of generating a clone sequence given an assembled fragment set is addressed in section 3.2 and the problem of generating an assembly given a clone sequence is addressed in section 3.3.

3.1.3 Likelihood

In this section we define the likelihood $\Pr(\mathbf{F} \mid \mathbf{A}, \Gamma, \mathrm{s}, \theta)$. We assume that each fragment f_j is an independent realization of the copying process, thus

$$\Pr(\mathbf{F} \mid \mathbf{A}, \Gamma, \mathrm{s}, \theta) = \prod_{j=1}^{m} \Pr(\mathrm{f}_j \mid \vec{\alpha_j}, \mathrm{s}, \theta).$$

This independence assumption is crucial to the analysis below. However, systematic errors, errors that recur at the same clone position in different fragments are known to occur and may account for a large proportion of the errors that find their way into finished reconstructions of the clone sequence. In section 4.3 below, we discuss this problem.

We can express $\Pr(\mathrm{f}_j \mid \vec{\alpha_j}, \mathrm{s}, \theta)$ as a product with one term for each arc in the alignment path because we have assumed a Markov model for the fragments generator. Let $v(i_t, k_t)$ be the vertex at which the partial alignment $\alpha_{j1}, \ldots, \alpha_{jt}$ terminates. Then

$$\Pr(\mathrm{f}_j \mid \vec{\alpha_j}, \mathrm{s}, \theta) = \prod_{t=1}^{n} \Pr(f_{jk_t} \mid \alpha_{jt}, s_{i_t}, \theta)$$

where,

$$\Pr(f_{jk_t} \mid \alpha_{jt}, s_{i_t}, \theta) = \begin{cases} 1 & \alpha_{jt} = 0 \quad \text{left end} \\ 1 & \alpha_{jt} = 1 \quad \text{first "one"} \\ \pi_R(-|s_{i_t}) & \alpha_{jt} = 0 \quad \text{interior} \\ \pi_R(f_{jk_t}|s_{i_t}) & \alpha_{jt} = 1 \quad \text{interior} \\ \pi_I(f_{jk_t}) & \alpha_{jt} = 2 \quad \text{interior} \\ 1 & \alpha_{jt} = 1 \quad \text{last "one"} \\ 1 & \alpha_{jt} = 0 \quad \text{right end.} \end{cases}$$

To achieve our primary goal of restoring the clone sequence s, we would like to know the assembled fragments and the correspondence vector. Thus ideally we would like to augment $\mathbf{F}$ with the "missing data" $\{\mathbf{A}, \Gamma\}$ and consider the augmented data likelihood. Our assumption of independent fragments implies the factoring

$$\Pr(\mathbf{F}, \mathbf{A}, \Gamma \mid \mathrm{s}, \theta) = \prod_{j=1}^{m} \Pr(\mathrm{f}_j, \vec{\alpha_j} \mid \mathrm{s}, \theta).$$

Let i_t and k_t be defined as before, then

$$\Pr(\mathrm{f}_j, \vec{\alpha_j} \mid \mathrm{s}, \theta) = \prod_{t=1}^{n} \Pr(f_{jk_t}, \alpha_{jt} \mid s_{i_t}, \theta)$$

where,

$$\Pr(f_{jk_t}, \alpha_{jt} \mid s_{i_t}, \theta) = \begin{cases} 1-\tau & \alpha_{jt}=0 \quad \text{left end} \\ \tau & \alpha_{jt}=1 \quad \text{first "one"} \\ (1-\lambda)(1-\mu)\pi_R(-|s_{i_t}) & \alpha_{jt}=0 \quad \text{interior} \\ (1-\lambda)(1-\mu)\pi_R(f_{jk_t}|s_{i_t}) & \alpha_{jt}=1 \quad \text{interior} \\ \lambda(1-\mu)\pi_I(f_{jk_t}) & \alpha_{jt}=2 \quad \text{interior} \\ \mu & \alpha_{jt}=1 \quad \text{last "one"} \\ 1 & \alpha_{jt}=0 \quad \text{right end.} \end{cases}$$

Prior and posterior distributions for the parameters λ, τ, μ, π_R and π_I are discussed in section 3.4.

3.1.4 A Sampling Algorithm

The primary objective of a sequencing project is to obtain a restoration of the clone sequence using information in the fragment sequences, prior information about the clone sequence (e.g., its length and base composition), and prior information about the frequencies and types of errors that occur in fragment sequences. We also wish to quantify any uncertainty in the restoration. For these purposes we would like to compute the marginal posterior distribution $\Pr(\mathrm{s} \mid \mathbf{F})$. The Gibbs sampling algorithm outlined here can be used to obtain an approximation to this distribution or functionals of it. For an introduction to the Gibbs sampler see Casella and George (1991). For theoretical properties and examples see Gelfand and Smith (1990) and Gelfand et al. (1991).

Our goal is to avoid both the assumptions that the assembled fragments matrix and the error rate parameters are fixed and known. Thus our goal is to estimate s from its distribution, marginal over $\mathbf{A}$ and θ. Any of several variations on Monte Carlo Markov chain algorithms could be used as a tool to solve this problem. We describe one that seems promising.

Starting with an initial assembly $\mathbf{A}^{(0)}$ and initial parameter estimates $\theta^{(0)}$ we iteratively generate the following random variables:

1. $\{\mathrm{s}, \Gamma\}^{(j)} \sim \Pr\left(\mathrm{s}, \Gamma \mid \mathbf{F}, \mathbf{A}^{(j-1)}, \theta^{(j-1)}\right)$,
2. $\theta^{(j)} \sim \Pr\left(\theta \mid \mathbf{F}, \{\mathrm{s}, \Gamma\}^{(j)}, \mathbf{A}^{(j-1)}\right)$,
3. $\mathbf{A}^{(j)} \sim \Pr\left(\mathbf{A} \mid \mathbf{F}, \mathrm{s}^{(j)}, \theta^{(j)}\right)$

Algorithms for each of these samplings steps are described in sections 3.2, 3.3 and 3.4 respectively.

Note that $\Gamma^{(j)}$ is discarded after step 2 of the iteration. Thus a sequence (of sequences) $\mathrm{s}^{(1)}, \ldots, \mathrm{s}^{(k)}$ is generated, where $\mathrm{s}^{(k)}$ is approximately a sample from $\Pr(\mathrm{s} \mid \mathbf{F})$. The approximation improves as k increases and becomes exact as $k \to \infty$.

This scheme generates a Markov chain $\left\{\mathrm{s}^{(j)}, \mathbf{A}^{(j)}, \theta^{(j)}\right\}$ with stationary distribution $\Pr(\mathrm{s}, \mathbf{A}, \theta \mid \mathbf{F})$. We may repeat the entire process N times or sample N

outcomes from one long chain (Gelman and Rubin 1992. Geyer 1992) to obtain the values

$$\begin{aligned} &\mathrm{s}_1^{(k)}, \mathrm{s}_2^{(k)}, \ldots, \mathrm{s}_N^{(k)} \\ &\theta_1^{(k)}, \theta_2^{(k)}, \ldots, \theta_N^{(k)} \\ &\mathbf{A}_1^{(k)}, \mathbf{A}_2^{(k)}, \ldots, \mathbf{A}_N^{(k)}. \end{aligned} \tag{4}$$

For large k we can treat these as a sample from the stationary distribution.

Using this sample to estimate probabilities is straightforward. For example, to estimate the probability that the clone sequence is a particular value s_0, we calculate

$$\Pr(\mathrm{s} = \mathrm{s}_0 \mid \mathbf{F}) \approx \frac{1}{N} \sum l = 1^N 1(\mathrm{s}_i^{(l)} = \mathrm{s}_0). \tag{5}$$

It has been noted (see for example Gelfand and Smith, 1990) that the estimate in (5) can be improved by applying the Rao–Blackwell theorem, which results in the estimator

$$\Pr(\mathrm{s} = \mathrm{s}_0 \mid \mathbf{F}) \approx \frac{1}{N} \sum_{l=1}^{N} \Pr\left(\mathrm{s} = \mathrm{s}_0 \quad \mathbf{F}, \mathbf{A}^{(l)}, \Gamma^{(l)}, \theta^{(l)}\right). \tag{6}$$

Lastly, we again note two properties of these calculations. The expression in (6) becomes exact as k and $N \to \infty$. Thus by taking large enough values, we can attain any degree of accuracy in these calculations. Also, the calculation produces a probability that does depend on any estimated values of θ. Similarly, calculations about θ do not depend on any estimated values of s.

In the remainder of this section, we provide the details of sampling from the three conditional distributions.

3.2 Conditional posterior distribution of the clone sequence

In this section we describe the conditional posterior distribution of the clone sequence $\Pr(\mathrm{s}, \Gamma \mid \mathbf{F}, \mathbf{A}, \theta)$ and an algorithm that generates samples from it.

Let s_i^* denote the subsequence of clone bases associated with the ith column of the assembly. The number of bases in s_i^* is given by γ_i. Recall the notation for the assembled fragments matrix, $\mathbf{X} = \{\mathbf{F}, \mathbf{A}\}$. It follows from the copying model that we have

1. Mutual independence of the (s_i^*, γ_i) given X.
2. Independence of x_i and $\left(s_j^*, \gamma_i\right)$ for $i \neq j$.

Thus the posterior distribution can be factored

$$\Pr(\mathrm{s}, \Gamma \mid \mathbf{X}, \theta) = \prod_{i=1}^{n_A} \Pr(s_i^*, \gamma_i \mid \mathrm{x}_i, \theta). \tag{7}$$

and we can restore s_i^* independently for each column.

The conditional distribution required to generate $s_i^\star$ can be computed using Bayes' rule

$$\Pr(s_i^\star, \gamma_i \mid \mathbf{x}_i, \theta) \propto \Pr(s_i^\star \mid \theta) \prod_{j=1}^{m} \Pr(x_{ij} \mid s_i^\star, \gamma_i, \theta). \tag{8}$$

The conditional probabilities of fragment elements given the clone sequence are

$$\begin{aligned} \Pr(x_{ij} \mid \gamma_i = 0) &= \begin{cases} \lambda \pi_I(x_{ij}) & x_{ij} \in \mathcal{B} \\ 1 - \lambda & x_{ij} = - \end{cases} \\ \Pr(x_{ij} \mid s_i^\star = b, \gamma_i = k, k \geq 1) &= \pi_R(x_{ij} | b_1) \prod_{l=2}^{k} \pi_R(- | b_l) \end{aligned}$$

where $b \in \mathcal{A}^k$.

The prior distribution for $(\mathbf{s}, \Gamma)$ will be defined in two stages. First we define a prior on Γ, the number of bases associated with each column in the assembly. Then, given the "size class" γ_i, we define a prior on the bases in $s_i^\star$. We will assume independence across columns of the assembly. Thus,

$$\begin{aligned} \Pr(\mathbf{s}, \Gamma) &= \prod_{i=1}^{n_A} \Pr(s_i^\star, \gamma_i) \\ &= \prod_{i=1}^{n_A} \Pr(\gamma_i) \Pr(s_i^\star \mid \gamma_i). \end{aligned}$$

Given that $s_i^\star$ belongs to a particular size class, the prior distribution will be equally likely,

$$\Pr(s_i^\star = b \mid \gamma_i = k) = \frac{1}{4^k} \tag{9}$$

for all $b \in \mathcal{A}^k$ and $k = 0, 1, 2, \ldots$.

To define a the prior on Γ, let $\eta_0 = \Pr(\gamma_i = 0)$ and distribute the remaining probability mass over the size classes $k = 1, 2, \ldots$ according to a geometric distribution with parameter η_1. Thus,

$$\Pr(\gamma_i = k) = (1 - \eta_0)(1 - \eta_1)\eta_1^{k-1}, \tag{10}$$

for $k \geq 1$. The prior expected length of $\mathbf{s}$ is

$$\mathrm{E}(n_s) = n_A \frac{1 - \eta_0}{1 - \eta_1}. \tag{11}$$

3.3 Conditional posterior distribution of the error rate parameters

In this section, we assume that the clone sequence and its correspondence with the assembled fragment set are known and consider the problem of estimating the error rate parameters. We will describe the conditional distribution $\Pr(\theta \mid \mathbf{F}, \mathbf{A}, \Gamma, \mathbf{s})$.

Prior Distribution The copying model is defined in terms of two sets of parameters, the state transition parameters λ, τ and μ and the output parameters $\pi_R(b|a)$ and $\pi_I(b)$. It is convenient to assign a beta prior distribution to λ with parameters β_λ and $\beta_{1-\lambda}$. The parameters τ and μ can also be treated this way, but see the discussion in section 4.

In general $\pi_R(b|a)$ is a 4×6 stochastic matrix with row sums equal to one and $\pi_I(b)$ is a 5 element probability vector. It is convenient to assign independent Dirichlet priors to each row of π_R and to π_I with parameters β^R_{ab} and β^I_b, respectively.

Posterior Distribution When $\mathbf{s}$, $\mathbf{F}$, $\mathbf{A}$ and Γ are given it is a simple counting excercise to determine which events have occurred in the process of copying all the fragments. The posterior distribution will be a product of Dirichlet distributions with parameters

$$\beta^\star_i = \beta_i + t_i \tag{12}$$

where t_i are the sufficient statistics

$$\begin{aligned}
t^R_{ab} &= \sum_{i=1}^{n_A}\sum_{j=1}^{m} \mathbf{1}(x_{ij} = b, s_{k(i)} = a),\ a \in \mathcal{A}, b \in \mathcal{B} \\
t^R_{a-} &= \sum_{i=1}^{n_A}\sum_{j=1}^{m} \left(\mathbf{1}(x_{ij} = -, s_{k(i)} = a) + \mathbf{1}(\gamma_i > 1)(\gamma_i - 1)d_i\right),\ a \in \mathcal{A} \\
t^I_b &= \sum_{i=1}^{n_A}\sum_{j=1}^{m} \mathbf{1}(x_{ij} = b, s_{k(i)} = -),\ b \in \mathcal{B} \\
t_\lambda &= \sum_{i=1}^{n_A}\sum_{j=1}^{m} \mathbf{1}(x_{ij} \in \mathcal{B}, \gamma_i = 0) \\
t_{1-\lambda} &= \sum_{i=1}^{n_A}\sum_{j=1}^{m} \mathbf{1}(x_{ij} \in \mathcal{B} \bigcup \{-\})\gamma_i
\end{aligned}$$

where $k(i)$ is the index of the clone base corresponding to column i in the assembly and d_i is the number of non–null characters in column i. Samples can be drawn from the posterior distributions using standard methods.

3.4 Conditional posterior distribution of the alignments

3.4.1 Approach

The alignment of DNA sequences is a ubiquitous problem in molecular biology (see the review by Waterman, 1984). In the study of molecular evolution, alignments are used to establish a correspondence among the bases in two or more related sequences that reflects their descent from a common base in an ancestral sequence. In the context of DNA sequencing, we can view the fragment sequences as descendents of the clone sequence via the copying process. An alignment between a fragment sequence $\mathbf{f}_j$ and a clone sequence $\mathbf{s}$ will establish which bases in the fragment were copied from which bases in the clone. In this

section we will describe the conditional posterior distribution of a pairwise sequence alignment $\Pr(\vec{\alpha}_j \mid \mathrm{s}, \mathrm{f}_j, \theta)$ and an algorithm that will generate samples from this distribution. The set of sampled pairwise alignments (one for each fragment, $j = 1, \ldots, m$) can then be used to construct an assembly drawn from the distribution $\Pr(\mathbf{A}, \Gamma \mid \mathbf{F}, \mathrm{s}, \theta)$. The problem of sampling alignments in an evolutionary context is addressed by Churchill and Thorne (1993) and a related EM algorithm is described by Thorne and Churchill (1993).

In general, the alignment of multiple sequences is a computationally prohibitive problem (Altschul, 1989). However, in the present case, the complexity is greatly reduced because (1) the common ancestral sequence s is given and (2) the fragment sequences are conditionally independent given s. Thus the joint distribution of multiple sequence alignment can be factored

$$\begin{aligned} \Pr(\mathbf{A}, \Gamma \mid \mathbf{F}, \mathrm{s}, \theta) &= \Pr(\vec{\alpha}_1, \ldots, \vec{\alpha}_m \mid \mathbf{F}, \mathrm{s}, \theta) \\ &= \prod_{j=1}^{m} \Pr(\alpha_j \mid \mathrm{f}_j, \mathrm{s}, \theta) \end{aligned}$$

and we can sample from the joints distribution of alignments by sampling the pairwise alignments one at a time.

For the remainder of section 3.4, we will consider the pairwise alignment distribution $\Pr(\vec{\alpha} \mid \mathrm{f}, \mathrm{s}, \theta)$ for a single fragment sequence f and the fragment subscript $j = 1, \ldots, m$ will be suppressed.

A partial alignment is an alignment between subsequences of two larger sequences. We will use the notation $\mathcal{A}(k, i, j)$ to denote the set of all partial alignments between $s_1, \ldots, s_i$ and $f_1, \ldots, f_j$ that end with an arc of type k. We refer to these sets as arc–sets. The following arc–sets are all non–empty:

1. Alignments that start after s_i:

$$\mathcal{A}(0, i, -1),\ i = 0, \ldots, n_s - 1$$

2. Alignments that start at s_i:

$$\mathcal{A}(1, i, 0),\ i = 0, \ldots, n_s$$

3. Alignments that enter node $v(i, j)$ by a k–path:

$$\begin{aligned} \mathcal{A}(k, i, j), \quad & k = 0,\ i = 0, \ldots, n,\ j = 1, \ldots, n_f \\ & k = 1,\ i = 1, \ldots, n,\ j = 1, \ldots, n_f \\ & k = 2,\ i = 1, \ldots, n,\ j = 0, \ldots, n_f \end{aligned}$$

4. Alignments that end at s_{i-1}:

$$\mathcal{A}(1, i, n_f + 1),\ i = 1, \ldots, n + 1$$

5. Alignments that end before s_{i-1}:

$$\mathcal{A}(0, i, n_f + 1),\ i = 2, \ldots, n + 1$$

All other arc–sets are null and should be assigned probability zero in the recursions below.

3.4.2 Forward pass algorithm

Define $q_k(i,j)$ to be the conditional probability that an arc of type k enters $v(i,j)$ given that $v(i,j)$ is visited by the alignment path. The goal of the forward pass algorithm is to compute $q_k(i,j)$ for all non–null arc–sets. First note that

$$\begin{aligned} q_k(i,j) &= \Pr\left(\mathcal{A}(k,i,j) \mid \mathcal{A}(\cdot,i,j), \mathbf{f}, \mathbf{s}, \theta\right) \\ &= \Pr\left(\mathcal{A}(k,i,j) \mid \mathcal{A}(\cdot,i,j), f_1, \ldots, f_j, \mathbf{s}, \theta\right) \\ &= \frac{\Pr\left(\mathcal{A}(k,i,j), f_1, \ldots, f_j \mid \mathbf{s}, \theta\right)}{\sum_{m=0}^{2} \Pr\left(\mathcal{A}(m,i,j), f_1, \ldots, f_j \mid \mathbf{s}, \theta\right)} \\ &= \frac{r_k(i,j)}{\sum_{k=0}^{2} r_k(i,j)} \end{aligned}$$

where

$$r_k(i,j) \equiv \Pr\left(\mathcal{A}(k,i,j), f_1, \ldots, f_j \mid \mathbf{s}, \theta\right)$$

and and

$$\mathcal{A}(\cdot,i,j) = \bigcup_{k=0}^{2} \mathcal{A}(k,i,j).$$

Claim: The following recursion computes $r_k(i,j)$ for all well defined arc sets.

1. start–point arc sets

$$\begin{aligned} r_2(0,-1) &= 1-\tau \\ r_2(i,-1) &= (1-\tau) r_2(i-1,-1),\ i = 1,\ldots,n_s-1 \\ r_1(0,0) &= \tau \\ r_1(i,0) &= \tau r_2(i-1,-1),\ i = 1,\ldots,n_s \end{aligned}$$

2. interior arc sets

$$\begin{aligned} r_2(i,j) &= (1-\lambda)(1-\mu)\pi_R(-|s_i) \sum_{m=0}^{2} r_m(i-1,j),\ i = 1,\ldots,n_s,\ j = 0,\ldots,n_f \\ r_1(i,j) &= (1-\lambda)(1-\mu)\pi_R(f_j|s_i) \sum_{m=0}^{2} r_m(i-1,j-1),\ i = 1,\ldots,n_s,\ j = 1,\ldots,n_f \\ r_0(i,j) &= \lambda(1-\mu)\pi_I(f_j) \sum_{m=0}^{2} r_m(i,j-1),\ i = 1,\ldots,n_s-1,\ j = 1,\ldots,n_f \\ r_0(n_s,j) &= \lambda\pi_I(f_j) \sum_{m=0}^{2} r_m(n_s-1,j-1),\ j = 1,\ldots,n_f \end{aligned}$$

3. terminal arc sets

$$\begin{aligned} r_1(i, n_f+1) &= \mu \sum_{m=0}^{2} r_m(i-1, n_f),\ i = 1, \ldots, n_s \\ r_1(n_s+1, n_f+1) &= (1-\lambda) \sum_{m=0}^{2} r_m(n_s, n_f) \\ r_2(2, n_f+1) &= r_1(1, n_f+1) \\ r_2(i, n_f+1) &= r_1(i-1, n_f+1) + r_2(i-1, n_f+1),\ i = 3, \ldots, n+1 \end{aligned}$$

Proof: We will work out the case of $r_1(i,j)$ for interior arcs. Other cases are proved similarly. All probabilities are conditional on θ which is suppressed in the notation here.

$$\begin{aligned} r_1(i,j) &= \Pr\left(\mathcal{A}(1,i,j), f_1, \ldots, f_j \mid \mathrm{s}\right) \\ &= \sum_{m=0}^{2} \Pr\left(\mathcal{A}(m,i-1,j-1), \mathcal{A}(1,i,j), f_1, \ldots, f_j \mid \mathrm{s}\right) \\ &\quad \text{by law of total probability} \\ &= \sum_{m=0}^{2} \Pr\left(\mathcal{A}(m,i-1,j-1), f_1, \ldots, f_{j-1} \mid \mathrm{s}\right) \\ &\quad \times \Pr\left(\mathcal{A}(1,i,j), f_j \mid \mathcal{A}(m,i-1,j-1), f_1, \ldots, f_{j-1}, \mathrm{s}\right) \\ &\quad \text{by definition of conditional probability} \\ &= \sum_{m=0}^{2} r_m(i-1,j-1) \Pr\left(\mathcal{A}(1,i,j), f_j \mid \mathcal{A}(m,i-1,j-1), \mathrm{s}\right) \\ &\quad \text{by definition of } r_k(i,j) \text{ and a conditional independence assumption} \\ &= \sum_{m=0}^{2} r_m(i-1,j-1) \Pr\left(f_j \mid \mathcal{A}(1,i,j), \mathcal{A}(m,i-1,j-1), \mathrm{s}\right) \\ &\quad \times \Pr\left(\mathcal{A}(1,i,j) \mid \mathcal{A}(m,i-1,j-1), \mathrm{s}\right) \\ &\quad \text{by a conditional independence assumption} \\ &= \sum_{m=0}^{2} r_m(i-1,j-1) \Pr\left(f_j \mid \mathcal{A}(1,i,j), \mathrm{s}\right) \Pr\left(\mathcal{A}(1,i,j) \mid \mathcal{A}(m,i-1,j-1), \mathrm{s}\right) \\ &= \pi_R(f_j \mid s_i)(1-\lambda)(1-\mu) \sum_{m=0}^{2} r_m(i-1,j-1). \end{aligned}$$

3.4.3 Traceback

We can express the likelihood of an alignment as

$$\Pr\left(\vec{\alpha} \mid \mathrm{f}, \mathrm{s}, \theta\right) = \prod_{t=n}^{1} \Pr\left(\alpha_t \mid \alpha_{t+1}, \ldots, \alpha_n, \mathrm{f}, \mathrm{s}, \theta\right)$$

$$= \prod_{t=n}^{1} \Pr\left(\mathcal{A}(\alpha_t, i_t, j_t) \mid \mathcal{A}(\cdot, i_t, j_t), \mathrm{f}, \mathrm{s}, \theta\right)$$
$$= \prod_{t=n}^{1} q_{\alpha_t}(i_t, j_t)$$

where n is the length of the alignment. The second equality follows from a conditional independence assumption. Given that $v(i_t, j_t)$ is visited by the alignment path, α_t is independent of the of the particular path $\alpha_{t+1}, \ldots, \alpha_n$ that extends from $v(i_t, j_t)$.

Once the forward pass algorithm is complete and the quantities $q_k(i,j)$ have been computed for each arc set, we can resample alignment paths. The traceback algorithm begins at the terminal node $(n+1, n_f+1)$ and continues until the start node $(-1,-1)$ is reached. From the node (i,j), we choose an arc set $\mathcal{A}(k,i,j)$ at random (among the non–null arc sets available) with probability $q_k(i,j)$. If $k=0$ then the traceback moves to node $(i, j-1)$. If $k=1$ it moves to node $(i-1, j-1)$ and if $k=2$ it moves to node $(i-1.j)$. The probability that any particular path $\vec{\alpha}$ is generated is

$$\Pr\left(\vec{\alpha} \mid \mathrm{f}, \mathrm{s}, \theta\right) = \prod_{t=n}^{1} q_{k_t}(i_t, j_t)$$

where k_t, i_t and j_t are defined by the arcs in the sampled alignment path.

3.5 An Example

A small simulated example is described to illustrate the algorithm of section 3.1.4. See figure 6 for details.

The observed data are a set of ten fragment sequences,

$$\begin{aligned}
\mathrm{f}_1 &= CCNTAT \\
\mathrm{f}_2 &= ACCCC \\
\mathrm{f}_3 &= TATCC \\
\mathrm{f}_4 &= GATAAAT \\
\mathrm{f}_5 &= TGGA \\
\mathrm{f}_6 &= TGAA \\
\mathrm{f}_7 &= GTAAC \\
\mathrm{f}_8 &= CTAAN \\
\mathrm{f}_9 &= ATGT \\
\mathrm{f}_{10} &= TGGTA.
\end{aligned}$$

The initial guess at the clone sequence is $\mathrm{s}^{(0)} = ATCGTGTATCCCTATGAAT$. The model parameters are fixed throughout this example at the values used to simulate the data that is, we skip algorithm step 2 for the sake of simplicity. The state transition probabilities are $\tau = \lambda = \mu = 0.1$. The rates of incorrectly

copied bases are $\pi_R(i|j) = 0.033$ for $i \in \mathcal{B}, j \in \mathcal{A}, i \neq j$ and the deletion rate is $\pi_R(-|j) = 0.1$. Insertions are equally likely to generate any of the bases in $\mathcal{B}$.

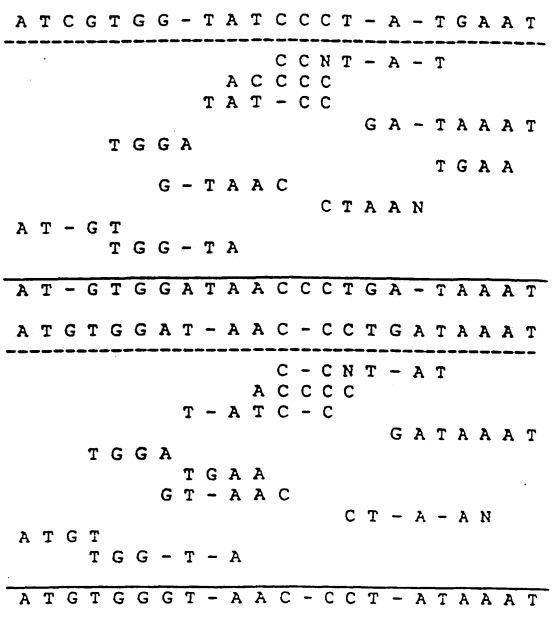

Figure 6: A small example of the Gibbs chain. From top to bottom we see: (i) the initial clone sequence $s^{(0)}$ shown with its $\gamma^{(1)}$ the assembly below it, (ii) an assembled fragments matrix $\mathbf{A}^{(1)}$ with 10 fragments, (iii) the sampled clone sequence $s^{(1)}$, $\gamma^{(1)}$ as inferred from the first assembly, (iv) the same sequence $s^{(1)}$ shown in its correspondence $\gamma^{(2)}$ with the second assembly (v) a second assembled fragments matrix $\mathbf{A}^{(1)}$, and (vi) the sampled clone sequence $s^{(2)}$, $\gamma^{(2)}$.

The initial assembly is generated (algorithm step 3) by aligning individual fragment sequences to the clone sequence $\mathrm{s}^{(0)}$ as shown at the top of figure 6 using the conditional distribution $\Pr(\mathbf{A}, \Gamma \mid \mathbf{F}, \mathrm{s}, \theta)$ given in section 3.4. Below the initial assembly in figure 6, the next clone sequence (algorithm step 1) $\mathrm{s}^{(1)} = ATGTGGATAACCCTGATAAT$ is shown in its correspondence ($\Gamma^{(1)}$ = 11011111111111111011111) with the initial assembly. $s^{(1)}$ and $\Gamma^{(1)}$ are generated from $\Pr(\mathrm{s}, \Gamma \mid \mathbf{F}, \mathbf{A}, \theta)$ given in section 3.2. The correspondence vector $\Gamma^{(1)}$ is discarded and a new alignment of the fragments is generated (algorithm step 3) as shown in the lower portion of figure 6. Another clone sequence is generated (algorithm step 1) $\mathrm{s}^{(2)} = ATGTGGGTAACCCTATAAAT$ and so on.

4 Extensions of the Copying Model

In this section, we discuss some of the experimental realities of sequencing and suggest how the HMM model of fragment generation described in section 3 could be extended to accomadate these.

4.1 Start–point Issues

The double–stranded structure of the DNA molecule introduces a complication into the assembly stage of a DNA sequencing project. The fragment sequences may be obtained as copies from either strand of the clone and it will generally not be known which fragments are copies of which strand. To account for this, we can extend the copying model of section 3.1 by adding a second hidden Markov chain that runs in the reverse direction along the clone sequence and generates copies of the complementary bases $s^c_n, \ldots, s^c_1$. We use the notation s^c to denote complementary bases, e.g. $A^c = T$. The states of the reversed Markov chain can be denoted by B^c_i, R^c_i, I^c_i and E^c_i. The start point of the copying process is modified so that $\Pr(B_0) = \Pr(B^c_0) = 1/2$. A fragment is equally likely to be generated as a direct or reversed complement copy of the clone.

The simple copying model implies that startpoints are (approximately) geometrically distributed along the length of the clone. However, a uniform distribution is probably more realistic. Furthermore we may wish to allow for the possibility that a fragment does not overlap the clone sequence at all. This will be convenient if the fragment sequences are being aligned to several unconnected segments of a clone sequence or if some fragments are expected to be "junk". Let the prior probability of no overlap be δ. If we now allow each state B_i to have its own transition probability τ_i, we can distribute the mass $1-\delta$ uniformly along the clone sequence by setting $\tau_i = \delta/(n_A - i + 1)$.

4.2 Fragment Dependent Errors

4.2.1 Error rates vary with position

Because the rate of migration of DNA molecules through a gel is non–linear, the ability to resolve bases is not constant across the length of a fragment. In particular the resolution decreases as the length of the gel read increases, resulting in more ambiguous (Ns) and less reliable base calls. In some systems there can also be resolution problems at the beginning of a gel read.

Koop et al. (1992) have reported a study of sequencing errors as a function of position along the gel on an automated fluorescent sequencer for two types of sequencing reactions. The general pattern of errors was found to be similar for both reactions and may be largely attributable to the nature of the gel and the base calling algorithms. They find that over the first 350 bases of the fragment, the error rate was roughly constant at about 1%. Beyond this point errors increased to about 17% at 500 bases. Deletions are the first type of error to increase starting at about base 350. They reach a peak of 3.5% at 400–450 bases and decrease thereafter. The next class of errors to increase (at 350–400 bases) are replacements and ambiguous base calls. These increase to about 8 to 10% at 500 bases. Insertions are the last type of error to increase, starting at about 450 bases. The insertion rate increases to greater than 10% at about 550–600 bases into the fragment sequence. Few fragments were available beyond 500 bases and these regions were difficult to align. Clearly more empirical studies of this type are needed to help us understand the error characteristics of raw

sequencing data and hence of the finished sequence.

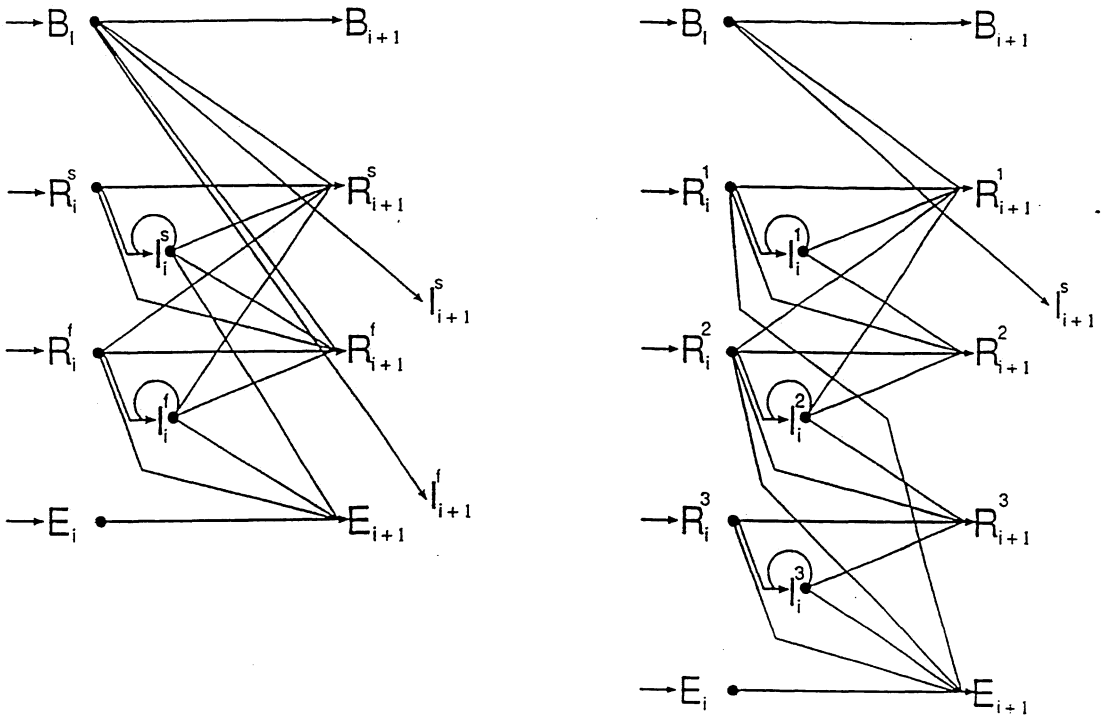

Figure 7: Hidden Markov models with non-uniform error rates across the bases in a fragment. The first model (a) has two interchanging sets of states, (R^s, I^s) and (R^f, I^f) corresponding to regions with low and high error rates in the fragment. The second (b) is a processive model with three sets of states (R^1, I^1), (R^2, I^2), (R^3, I^3) that correspond to early middle and late bases in the gel run.

These studies suggest that the uniform error rates model is inadequate to describe the process of fragment generation and may be misleading. A base determination at 100 bases into a fragment should be treated as being more reliable than a base determination at 450 bases into a fragment. Biologists involved in sequencing have been aware of this decay accuracy and will take it into account when ambiguities are resolved in the process of rechecking or constructing sequences "by hand". However when faced with the task of large scale and fully automated sequence assembly, we will need to develop an appropriate weighting scheme. Huang (1992) implemented a two–stage weighting scheme into his fragment assembly software. User defined cutoff points at the beginning and end of each fragment are used to identify extremal regions where error rates may exceed 5%.

The HMM copying model can be extended to include fragment position effects by allowing the parameters associated with R–states and I–states to depend on the fragment position. In a simple case we might allow two types of states R^f, I^f would be "fast" states with high error rates and R^s, I^s would be "slow" states with lower error rates. This model would be easy to implement and may be sufficient to solve the problem. Figure 7 illustrates the basic unit of the

HMM for this model and also for a three state model in which the states are progressive, i.e. start of the fragment, middle of the fragment and end of the fragment. A more general solution would allow the model parameters to depend on the fragment position, t. This essentially introduces an infinite array of R and I–states for every base in the clone with transitions restricted to be from t to $t+1$ in most cases. Empirical results could be used to develop reasonable prior distributions for position dependent error rates λ_t, $\pi_t(\cdot)$, and $\pi_t(\cdot|\cdot)$. The number of parameters involved is large and some smoothness constraints will be necessary (see the discussion by Roeder).

4.2.2 Error rates vary between fragments

Error rates are also known to vary between fragments. Some fragments are very reliable across a wide range of bases and others are more error–prone. It may be possible to extend the simple model to allow for fragment–to–fragment variation by including a fragment–specific parameter in a hierarchical model. It may also be possible to identify error–prone fragments and simply remove them from the assembly.

Occasionally in the process of fragmenting and subcloning DNA, two segments from different regions of the clone will be joined together. Such chimeric sequences can cause significant problems in assembly. If the assembly is correct they may appear to have high error rates in the misaligned portion. If a chimera causes the assembly to be incorrect, the whole region around the incorrect join may appear error prone. Identification of error prone and/or chimeric sequences remains an open problem of significant practical importance.

4.3 Sequence Dependent Errors

An assumption throughout this work has been that the fragments are independent realizations of the copying process. However, it becomes clear when looking at assembled sequence fragments, that the same errors sometimes tend to occur repeatedly at the same points in the clone sequence. One possible approach to this problem may be to allow the error rate parameters to vary with their position in the clone. Again a hierarchical model may be most appropriate. An approach similar to the multiple sequence alignment methods of Krogh et al. (1993) with the use of mixtures priors (Brown et al. 1993) to represent different error rate classes seems promising here.

One common source of errors is the miscalling of the length of a homopolymer run (e.g. TTTTTT). Another source of errors are compression, which may be related to sequence–specific structures that form in the DNA as it migrates through the gel. Compressions are most common in GC rich regions of the DNA and cause deletions and/or transpositions in the fragment sequences. The same errors will often occur in fragments sequenced in one orientation, but not those sequenced in the opposite orientation. Note that our model in section 3 does not allow for transposition as a class of errors, nor does it take account of the strand being copied. Again, these are issues that will have to be addressed as part of a practical solution to the DNA reconstruction problem.

5 Prospects for Coherent Estimation of DNA Sequences

If the goals of the human genome initiative are to be achieved within the next decade and at a reasonable cost, the process of DNA sequencing must become a large–scale production effort. New developments in sequencing technology are likely to arise and will facilitate this effort. However the tradeoffs between sequence quality and costs will surely be a factor. It is this author's opinion that it may be reasonable to sacrifice fidelity for speed in sequencing, provided that we can develop reliable statistical methods to interpret highly redundant, low–fidelity raw data and produce sequences with well–defined and acceptable error characteristics. Whatever methods are finally used to obtain large DNA sequences, it will be essential to develop reliable estimates of accuracy and to report the accuracy of each finished sequence using average and/or base–by–base measures.

The question of acceptable error rates is a matter of some vigorous debate among biologists due in part to the increasing cost of more accurate sequences. "Acceptable" error rates range from 0.05 per base to 0.0001 or less per base (for example, Hunkapillar et al 1991, States and Botstein 1991, Clark and Whittam 1993) and depend on the types of analyses for which the sequence is intended. We will not enter into this debate. However we do support the opinion of States (1992) that low–accuracy sequences can be a valuable resource provided the frequency and characteristics of errors are known. Thus there is a need for continued effort on the problem of estimating sequence accuracy, a problem the falls within the domain of statistics.

It is somewhat risky to write a methodology paper about DNA sequencing as the technology is constantly changing. However we can anticipate that, for at least the next several years, methods that produce sequence data as strings of contiguous characters, i.e. linear sequencing methods, will continue to be used. By incremental improvements, such as increasing the length of readable gels, the throughput of current technology can be improved several fold. However, if the goals of the human genome project are to be met within the next decade, it is likely that new high–speed technologies will be required.

As more efficient technologies and sequencing "tricks" are developed, it is likely that the shotgun approach to sequencing will be replaced by more directed strategies. With directed sequencing strategies, assembly is less problematic and the the redundancy of sequence determinations can be reduced. Chen et al. (1992) advocate this approach to large–scale sequencing. However, this will not eliminate the need for statistical analysis of the error properties of DNA sequence data. In fact, the opposite may be true. As the redundancy of coverage is reduced it will be necessary to assess the accuracy of sequences by relying more on our prior knowledge of the error characteristics of the system used. Prior information on a sequence production system could be gathered by repeated sequencing of known standards to establish its baseline error characteristics. As we discussed above, such an analysis should consider both (fragment) position effects and (clone) context effects on errors.

Finally we note that in the present work we have assumed the data are given as base calls on the alphabet $\{A, C, G, N, T\}$. In fact recent work on the base calling problem (Tibbets et al. 1993, Golden et al. 1993, Lawrence and Solovyev 1993) has focussed on the raw data streams (traces) generated by flourescence–based sequencing devices. Tibbets et al (1993) use neural networks and Lawrence and Solovyev use discriminant analysis methods to interpret these traces as mixtures of A, C, G, T and (in the work of Lawrence and Solovyev, 1993) undercall and overcall, thus effectively providing a probability distribution in place of the standard base call. It appears that direct utilization of traces can precisely identify most errors in the raw sequencing data. This approach also allows us to circumvent the problem of decay of accuracy along the length of a gel run as this decay is directly reflected in the probabilistic base calls.

Churchill and Waterman (1992) describe an approach to combining probabilistic base calls (assuming a fixed alignment) using Bayes' rule. The problem of combining probability distributions on sequence spaces (with the alignment assumption) remains unsolved. However extensions of the alignment and consensus estimation methods presented here may be obtained. We note the combination of trace data using Bayes' rule may be optimistic in that actual sequence traces are not necessarily independent realizations. Thus the error bounds may be optimistic in regions of high redundancy. Lower bounds on the accuracy might be obtained by taking the maximum probability over all traces or by combining the two maximal traces in regions where the sequence is obtained in both orientations.

We hope that the need for clear statistical thinking and in particular, Bayesian statistical thinking, as an essential component of a large–scale DNA sequencing project has been demonstrated. If statistical methods are to be successfully integrated into the sequencing process, they will have to be implemented in user–friendly and flexible software products. Such software should allow the scientist to assemble fragments, estimate a consensus sequence and assess the quality of the results within a unified and largely automated system. Direct intervention in the process should be possible when needed but the software should not require an expert statistician to run properly. Thus user input should be limited to a few critical parameters that are easily understood. Extensive prior information could be gathered automatically and accumulated in files without user intervention. Such a system should be capable of offering multiple solutions and (approximate) assessments of their reliability in the form of intuitive measures such as posterior probabilities.

References

Alizadeh, F., Karp, R.M., Newberg, L.A., Weisser, D.K. (1992) Physical mapping of chromosomes: A combinatorial problem in molecular biology. Preprint.

Altschul, S.F., Lipman, D.J. (1989) Trees, stars, and multiple biological sequence alignment. SIAM Journal on Applied Mathematics **49**:197-209.

Berger, J.O. (1985) *Statistical Decision Theory and Bayesian Analysis*. 2nd ed. Springer–Verlag.

Borodovsky, M. and McIninch, J. (1993a) Genmark: Parallel gene recognition for both DNA strands. Computers Chem. **17**:123-133.

Borodovsky, M. and McIninch, J. (1993b) Recognition of genes in DNA sequence with ambiguity. Biosystems **30**:161–171.

Bowling, J.M., Bruner, K.L., Cmarik, J.L., Tibbets, C. (1991) Neighboring nucleotide interactions during DNA sequencing gel electrophoresis. Nucl. Acids Res. **19**:3089–3097.

Branscomb, E. *et al.* (1990) Optimizing restriction fragment fingerprinting methods for ordering large genomic libraries. *Genomics* **8**:351-366.

Casella, G.C. and George, E.I. (1992) Explaining the Gibbs sampler American Statistician

Chen, E. *et al.* (1991) Sequence of the human glucose–6–phosphate dehydrogenase cloned in plasmids and a yeast artificial chromosome. *Genomics* **10**:792–800.

Chernoff H. (1992) Estimating a sequence from noisy copies. Harvard University technical report no. ONR-C-10.

Churchill, G.A. (1989) A stochastic model for heterogeneous DNA sequences. Bull. Math. Biol. **51**:79–94.

Churchill, G.A., Burks, C., Eggert, M., Engle, M.L., Waterman, M.S. (1992) Assembling DNA fragments by shuffling and simulated annealing. Manuscript.

Churchill, G.A. and Thorne, J.L. (1993) The probability distribution of a molecular sequence alignment. Cornell University, Biometrics Unit technical report.

Churchill, G.A. and Waterman, M.S. (1992). The accuracy of DNA sequences: estimating sequence quality. *Genomics* in press.

Clark, A.G. and Whittam T.S. (1992) Sequencing errors and molecular evolutionary analysis. Mol. Biol. Evol. **9**:744–752.

Clarke, L. and Carbon, J. (1976) A colony bank containing synthetic Col EI hybrid plasmids representative of the entire E. coli genome. Cell **9**:91–99.

Cornish–Bowden A. (1985) Nomenclature for incompletely specified bases in DNA sequences: Recommendations 1984. Nucl. Acids Res. **13**:3021–3030.

Daniels, D.L., Plunkett, G., Burland, V., Blattner, F.R. (1992) Analysis of the *Escherichia coli* genome: DNA sequence of the region from 84.5 to 86.5 minutes. *Science* **257**: 771–778.

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc. B **39**:1–38.

Edwards, A. *et al.* (1990) Automated DNA sequencing of the Human HPRT locus. *Genomics* **6**:593–608.

Fu, Y.-X., Timberlake, W.E., Arnold, J. (1992) On the design of genome mapping experiments using short synthetic oligonucleotides. Biometrics **48**:337–359.

Gelfand A.E. and Smith, A.F.M. (1990) Sampling based approaches to calculating marginal densities. J. Amer. Statist. Assoc. **85**:398–409.

Gelman, A. and Rubin, D.B. (1992) Inference from iterative simulation, with discussion. Statistical Science **7**:457–511.

Geyer, C.J. (1992) Markov chain Monte Carlo maximum likelihood. *Computer Science and Statistics: Proceeding of the 23rd symposium on the interface.*

Golden, J.B., Torgersen, D., Tibbets, C. (1993) Pattern recognition for automated DNA sequencing: I. On–line signal conditioning and feature extracttion for base-

calling. In *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology.* AAAI Press.

Hastings (1970) Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**:97–109.

Huang, X. (1992) A contig assembly program based on sensitive detection of fragment overlaps. *Genomics* **14**:18–25.

Hunkapillar, T, Kaiser, R.J., Koop, B.F., Hood, L. (1991) Large–scale automated DNA sequence determination. *Science* **254**:59–67.

Kececioglu, J. and Myers, E. (1990). A robust automatic fragment assembly system. Preprint.

Koop, B.F., Rowan, L., Chen, W.–Q., Deshpande, P., Lee, H. and Hood, L. (1993) Sequence length and error analysis of sequenase and automated *Taq* cycle sequencing methods. *BioTechniques* **14**:442–447.

Krawetz, S.A. (1989) Sequence errors described in GenBank: A means to determine the accuracy of DNA sequence interpretation. Nucl. Acids Res. **17**:3951–3957.

Krogh, A., Brown, M., Mian, I.S., Sjölander, K., Haussler, D. (1993) Hidden Markov models in computational biology: Applications to protein modeling. J. Mol. Biol., accepted.

Lander, E.S. and Waterman, M.S. (1988) Genomic mapping by fingerprinting random clones: A mathematical analysis. *Genomics* **2**:231–239.

Larson, S., Mudita, J., Myers, G. (1993) An interface for a fragment assembly kernal. University of Arizona, Department of Computer Science TR93-20.

Lawrence, C.B. and Solovyev, V.V. (1993) Assignment of position specific error probability to primary DNA sequence data. manuscript

Lewin, B. (1992) *Genes* V. Wiley, New York.

Maxam, A.M. and Gilbert, W. (1977) A new method for sequencing DNA. *Proc. Natl. Acad. Sci.* **74**:5463–5467.

Oliver, S.G., *et al.* (1992) The complete DNA sequence of yeast chromosome III. *Nature* **357**:38–46.

Posfai J. and Roberts R.J. (1992) Finding errors in DNA sequences. *Proc. Natl. Acad. Sci.* **89**: 4698–4702.

Roberts, L. (1990). Large-scale sequencing trials begin. *Science*, **250**: 1336–1338.

Sanger, F., Nicklen, S., and Coulson, A.R. (1977) DNA sequencing with chain terminating inhibitors. *Biochemistry* **74**:560–564.

Santner, T.J. and Duffy, D.E. (1989) *The Statistical Analysis of Discrete Data.* Springer–Verlag, NY.

Seto, D., Koop, B.F., Hood, L. (1993) An experimentally derived data set constructed for testing large–scale DNA sequence assembly algorithms. *Genomics* **15**:673–676.

Staden, R. (1980). A new computer method for the storage and manipulation of DNA gel reading data. Nucleic Acids Res. **8**:3673–2694.

States, D.J. (1992) Molecular sequence accuracy: analysing imperfect data. Trends in Genetics **8**:52–55.

States, D.J. and Botstein, D. (1991). Molecular sequence accuracy and the analysis of protein coding regions. Proc. Natl. Acad. Sci. USA **88**:5518–5522.

Sulston, J. em et al. (1992) The *C. elegans* genome sequencing project: a beginning. *Nature* **356**:37–41.

Thorne, J.L. and Churchill, G.A. (1993) Estimation and reliability of molecular sequence alignments. Biometrics, accepted.

Thorne JL, Kishino H, Felsenstein JF (1991) An evolutionary model for maximum likelihood alignment of DNA sequences. J. Mol. Evol. 33:114–124.

Thorne JL, Kishino H, Felsenstein JF (1992) Inching toward reality: An improved likelihood model of sequence evolution. J. Mol. Evol. 34:3–16.

Tibbets, C., Bowling, J.M., Golden, J.B. (1993) Neural networks for automated base calling of gel–based DNA sequencing ladders. In *Automated DNA Sequencing and Analysis Techniques* Dr. J. Craig Ventner, Editor, Academic Press.

Waterman MS (1984) General methods of sequence comparison. Bull. Math. Biol. 46:473–500.

Watson, J and Crick, F. (1953) Nature **171**: 737–738.

DISCUSSION

George Casella, Cornell University

Christian Robert, Cornell University

1 Introduction

The article by Professor Churchill provides a wonderful introduction to this fascinating subject. We heartily congratulate him. In this discussion we would like to examine two points in detail, with the joint goal of assessing the ability of the present methodology to produce a usable inferential procedure. First, we look in detail at implementing the Markov chain, both in computing the necessary distributions and generating the required random variables. Second, we outline a procedure for constructing a confidence set for the restored clone sequence. We then discuss the feasibility of implementing the algorithms.

2 Model and Notation

The Markov Chain model, as given by Churchill in Section 3.1.4, is

$$\begin{aligned} \{s,\Gamma\}^{(j)} &\sim \{s,\Gamma\}|F, A^{(j-1)}, \theta^{(j-1)} \\ \theta^{(j)} &\sim \theta|F, \{s,\Gamma\}^{(j)}, A^{(j-1)} \\ A^{(j)} &\sim A|F, s^{(j)}, \theta^{(j)} \end{aligned} \tag{1}$$

where

$\{s,\Gamma\}^{(j)} = s^{*(j)}$ = the restored clone sequence together with alignment information;

$$\begin{aligned} F &= \{f_1, \cdots, f_m\} \text{ are the m fragments;} \\ A^{(j)} &= \text{the assembly information for the fragments;} \\ \theta^{(j)} &= \{(\tau,\ \mu), (\lambda,\ \pi_I(\cdot)),\ \pi_R(\cdot|\cdot)\} \text{ are underlying parameters.} \end{aligned}$$

Together, F and A result in an $n_A \times m$ matrix $X = \{x_{ij}\}$, the assembled fragment set. This is the alignment of the fragments F according to the information in A. A few clarifying remarks are in order.

i) The difference between s and $\{s, \Gamma\}$ is subtle, but important. The clone sequence s, of length n_S, takes its values in $\{A, C, G, T\}$. The sequence $\{s, \Gamma\}$, of length n_A, takes its values in $\{-, A, C, G, T, \mathcal{A}^k, k = 2, 3, \cdots\}$ where $\mathcal{A}^k$ is made of all k-tuples of the $\{A, C, G, T\}$ alphabet. For example, we may have

$$s^* = \{s, \Gamma\} = |A|C| - |G^A|T| - |$$

of length $n_A = 6$, where G^A denotes the *insertion* of the element A at the fourth spot, and the corresponding s is $ACGAT$ of length $n_S = 5$. So $\{s, \Gamma\}$ is the clone sequence together with the alignment information, which is what is generated, while s is the inferred clone sequence, the object of interest.

ii) When generating the alignments $\vec{\alpha_1}, \cdots, \vec{\alpha_m}$ we, in fact, generate a new Γ for the clone sequence s. Thus, we may write the third step in the Markov chain (1) as

$$\{A, \Gamma^{(j)}\} \sim \{A, \Gamma\}|F, s^{(j)}, \theta^{(j)} \tag{2}$$

We could update the Γ part of $\{s, \Gamma\}^{(j)}$ at this point, or use only the $A^{(j)}$ from this generation.

iii) The groupings of parameters in θ reflect their purpose: (τ, μ) govern the beginning and ending of the copying process; $(\lambda, \pi_I(\cdot))$ govern the insertion process; and $\pi_R(\cdot|\cdot)$ governs the replacement process. In this discussion we will focus on a special case of π_I and π_R, but the mechanics of generalization are straightforward. We consider the special case

$$\begin{aligned} \pi_I(\cdot) &= \frac{1}{4} \\ \pi_R(b_1|b_2) &= p_S/4, \quad b_1 \in \mathcal{B},\ b_2 \in \mathcal{A},\ b_1 \neq b_2 \\ \pi_R(-|b) &= p_D, \quad b \in \mathcal{A} \\ \pi_R(b|b) &= 1 - p_S - p_D, \quad b \in \mathcal{A}. \end{aligned}$$

This is the case where all insertions are equally likely, as are all substitutions. To make these distinctions more clear, the following small example may be helpful.

Example: Suppose we have fragments $f_1 = AC$, $f_2 = GAT$, $f_3 = ACAT$.

- At step i, the generated clone sequence and alignment is

$$\{s, \Gamma\}_i = |A||G|| - |C|C^G|T| \tag{3}$$

(Note the C^G element is in $\mathcal{A}^2$.) The resulting generated clone sequence is s = AGCCGT. (The mechanics of mapping $\{s, \Gamma\}$ to s are to delete dashes and string out superscripts.)

• Given s and the three fragments, we generate three alignments $\vec{\alpha_1}$, $\vec{\alpha_2}$ and $\vec{\alpha_3}$

$$\begin{array}{cccccc|cccccc|cccccc}
 & & \multicolumn{4}{c}{\{f_1,\vec{\alpha_1}\}} & & & \multicolumn{4}{c}{\{f_2,\vec{\alpha_2}\}} & & & \multicolumn{4}{c}{\{f_3,\vec{\alpha_3}\}} & \\
A & G & C & C & G & T & A & G & C & C & G & T & A & G & C & C & G & T \\
A & - & C & - & - & - & - & - & - & - & G^A & T & A & - & C & - & A & T
\end{array} \tag{4}$$

These alignments result in $\{A, \Gamma\}_i$, the assembly and alignment information, and the assembled fragments matrix X_i, where

$$\begin{aligned}
\{A,\Gamma\}_i &= \begin{bmatrix} A & - & C & - & - & - & - \\ - & - & - & - & G & A & T \\ A & - & C & - & - & A & T \end{bmatrix}, \\
X_i &= \begin{bmatrix} A & C & - & - & - \\ - & - & G & A & T \\ A & C & - & A & T \end{bmatrix}
\end{aligned} \tag{5}$$

(The mechanics of $\{A, \Gamma\} \rightarrow X$ are to delete columns that contain only dashes.)

• Given the assembled fragments matrix X_i, we generate a new realization of $\{s, \Gamma\}$. The generation is done on a column by column basis:

$$X_i = \begin{array}{c} \begin{array}{ccccc} A^T & C & - & A & T \end{array} \\ \begin{bmatrix} A & C & - & - & - \\ - & - & G & A & T \\ A & C & - & A & T \end{bmatrix} \end{array}$$

yielding the new $\{s, \Gamma\} = |A^T|C|-|A|T|$. Note that A^T is generated because $\gamma_1 = 2$. When run to equilibrium, the output from the Gibbs sampler is a sample $\{s, \Gamma\}_i$, $i = 1, \cdots, k$ from the marginal distribution. In the next section we look at all of the steps of the chain in detail, and examine exactly how the needed densities are calculated and the random variables are generated.

3 Calculation and Generation of the Chain

There are three parts to the generation of the Markov Chain in (1), and each part presents its own difficulties. We will treat them in order.

It first will be useful to discuss various groupings of parameters and statistics, and the forms of these that are easiest to work with. The collection of fragments, F, represent the data (or, in EM algorithm terms, the "incomplete" data). A good choice of the "complete" data is $X = \{F, A\}$, the assembled fragments matrix. With this data, the parameter (clone) vector $s^* = \{s, \Gamma\}$ is

most straightforward to work with. Thus, we implement the Markov Chain (1) as

$$\begin{aligned} s^{*(j)} &\sim s^*|X^{(j-1)}, \theta^{(j-1)} \\ \theta^{(j)} &\sim \theta|X^{(j-1)}, s^{*(j-1)} \\ A^{(j)} &\sim A|F, s^{(j)}, \theta^{(j)}. \end{aligned} \tag{6}$$

Note that in the third step of (6) we only condition on $s^{(j)}$ from $s^{*(j)} = \{s, \Gamma\}^{(j)}$, and we actually generate a new Γ, which is discarded. The assembly $A^{(j)}$ is then used to update $X^{(j-1)}$ to $X^{(j)}$. In deriving the necessary posterior distributions we will work with the complete data likelihoods expressed in terms of X, as given by Churchill in Section 3.2. This strategy is easier to implement than working with the likelihoods based on F.

3.1 The $\{s, \Gamma\}$ distribution

Recalling that $X = \{F, A\}$, the desired distribution for the first part of the Markov chain in (1) is $\{s, \Gamma\}|\ X, \theta$. The matrix X is $m \times n_A$, and the vector $\{s, \Gamma\}$ is $1 \times n_A$. An element of $\{s, \Gamma\}$, say $\{s, \Gamma\}_i$, takes its values in $\{-\cup \mathcal{A}^k,\ k = 1, 2, \cdots\}$ and we will write either $\{s, \Gamma\}_i = -$ or $\{s, \Gamma\}_i = b \times b_{k-1}$. It is important to separate the first element in the k-tuple, as this is the only base for which the corresponding x_{ij} imparts any information. Now, using the specification of the prior and sampling distribution given by Churchill in Section 3.2. and using the fact that the columns of X are assumed independent, straightforward calculation yields the posterior distribution

$$\begin{aligned} P(\{s, \Gamma\}_i = b \times b_{k-1}|\mathbf{x}_i, \theta) &\propto (1 - p_S - p_D)^{\delta_i(b)} \left(\frac{1}{3}p_S + p_D\right)^{m-\delta_i(b)} \\ &\quad \times \frac{(1-\eta_0)(1-\eta_1)\eta_1{}^{k-1}}{4^k} \\ P(\{s, \Gamma\}_i = -|\mathbf{x}_i, \theta) &\propto \left(\frac{1}{4}\lambda^{T_i}\right)(1-\lambda)^{m-T_i}\,\eta_0, \end{aligned}$$

where $\mathbf{x}_i = i^{th}$ column of X, $\delta_i(b) = \sum_{j=1}^m I(x_{ij} = b)$ and $T_i = \delta_i(A) + \delta_i(C) + \delta_i(G) + \delta_i(T)$. The normalizing constant is simple to compute, being a sum of the geometric series, hence available in closed form. Finally, the full posterior of $\{s, \Gamma\}$ is, by independence,

$$P(\{s, \Gamma\}|\mathbf{X}, \theta) = \prod_{i=1}^{n_A} P(\{s, \Gamma\}_i|\mathbf{x}_i, \theta) \tag{7}$$

Generation of $\{s, \Gamma\}_i$ is, perhaps, most easily accomplished by first generating Γ_i (the "depth" of the element $\{s, \Gamma\}_i$), and then generating $\{s, \Gamma\}_i|\Gamma_i$. We do this using, from (6),

$$
\begin{aligned}
P(\Gamma_i = 0|\mathbf{x}_i, \theta) &\propto \left(\frac{1}{4}\lambda\right)^{T_i}(1-\lambda)^{m-T_i}\,\eta_0 \\
P(\Gamma_i = k|\mathbf{x}_i, \theta) &\propto (1-\eta_0)\,(1-\eta_1)^{\frac{\eta_1{}^{k-1}}{4^k}} \\
&\times \sum_{b\in\{A,C,G,T\}} (1-p_S-p_D)^{\delta_i(b)}\left(\frac{1}{3}p_S+p_D\right)^{m-\delta_i(b)}
\end{aligned}
$$

and

$$
\begin{aligned}
P(\{s,\Gamma\}_i = -|\Gamma_i = 0, \mathbf{x}_i, \theta) &= 1 \\
P(\{s,\Gamma\}_i = b\times b_{k-1}|\Gamma_i = k, \mathbf{x}_i, \theta) &\propto (1-p_S-p_D)^{\delta_i(b)}\left(\frac{1}{3}p_S+p_D\right)^{m-\delta_i(b)}\frac{1}{4^{k-1}}.
\end{aligned}
$$

Thus, Γ_i is generated using a geometric distribution, and then $\{s,\Gamma\}_i$ is a straightforward discrete generation. The necessary normalizing constants can also be easily calculated.

3.2 The Distribution of θ

The parameter vector $\theta = \{(\tau,\mu), \lambda, (p_S, p_D)\}$ plays no role in the ultimate inference on s, but provides an essential intermediate step in the model. Fortunately, calculation of the posterior distribution and generation of random variables is straightforward. We again start from the likelihood in terms of X, given by Churchill in Section 3.2. Recalling that $\{s,\Gamma\} = s^*$ and $\{F,A\} = X$, the posterior distribution of θ in (1) is

$$\pi(\theta|F,\{s,\Gamma\},A) = \pi(\theta|X,s^*) \propto P(X|s^*,\theta)P(s^*|\theta)\pi(\theta) \tag{8}$$

and, since the elements of X are independent given s^*, we can write

$$\pi(\theta|X,s^*) \propto \prod_{i=1}^{n_A}\prod_{j=1}^{m} P(x_{ij}|s_i^*,\theta)\,P(s_i^*|\theta)\,\pi(\theta). \tag{9}$$

Using the distributions given by Churchill in Section 3.2, and defining

$$
\begin{aligned}
\delta_{ij}(u,v) &= \begin{cases} 1 & \text{if } x_{ij} = u \text{ and } s_i^* = v \\ 0 & \text{otherwise,} \end{cases} \\
N_i^{(B)} &= \text{number of } \phi s \text{ in } x_{ij} \text{ before copying begins.} \\
N_i^{(E)} &= \text{number of } \phi s \text{ in } x_{ij} \text{ after copying ends.}
\end{aligned}
$$

we have

$$\pi(\theta|X,s^*) \propto [(1-\lambda)\eta_0]^{\sum_{ij}\delta_{ij}(-,-)}\prod_{k=1}^{\infty}\left[p_D(1-\eta_0)(1-\eta_1)\eta_1^{k-1}\right]^{\sum_{ij}\delta_{ij}(-,b_k)}$$

$$\times \prod_{l=1}^{4} [\lambda\gamma_0/4]^{\sum_{ij}\delta_{ij}(b_1^{(l)},-)} \tag{10}$$

$$\times \prod_{l=1}^{4}\prod_{k=1}^{\infty}\left[\frac{1-p_S-p_D}{4}(1-\eta_0)(1-\eta_1)\eta^{k-2_1}\right]^{\sum_{ij}\delta_{ij}(b_1^{(l)},b_1^{(l)}b_{k-1})}$$

$$\times \prod_{l=1}^{4}\prod_{l'=1 l\neq l'}^{4}\left[\frac{p_S}{4}(1-\eta_0)(1-\eta_1)\eta_1^{k-2}\right]^{\sum_{ij}\delta_{ij}(b_1^{(l)},b_1^{(l')}b_{k-1})}$$

$$\times \tau\mu(1-\tau)^{\sum_i N_i^{(B)}}(1-\mu)^{\sum_i N_i^{(E)}} \times \pi(\theta)$$

Although the products in (10) are infinite products, in practice they contain only n_A terms. This is because if s_i^* has depth d, then $\delta(u, v)$ is zero unless v also has depth d. If we then take $\pi(\theta)$ to be a product of a Dirichlet distribution on p_S and p_D, and independent beta distributions on λ, τ, μ, η_0 and η_1, the posterior distribution is again a product of a Dirichlet and independent beta distributions. Note that our $\delta_{ij}(\cdot,\cdot)$ notation is a more explicit form of Churchill's notation in Section 3.4. The δ_{ij} notation involves X and s^*, while the t_{ab} notation involves X and s. Of course, we could have written $\pi(\theta|X, s^*)$ in terms of different component distributions, in particular using the likelihood on F given by Churchill in Section 3.1.3. This would lead to

$$\pi(\theta|F,\{s,\Gamma\},A) \propto P(F,\vec{\alpha}|s,\theta)P(s,\theta) = P(F|\vec{\alpha},s,\theta)P(\vec{\alpha}|s,\theta)P(s|\theta)\pi(\theta)$$

where we have used the fact that $\{F,\vec{\alpha}\} = \{F,A,\Gamma\}$ However, this form of the posterior distribution seems much more difficult to work with. In particular, the distribution $P(s|\theta)$ is quite involved.

3.3 The Alignment Distribution

The third part of the Markov chain (1), the distribution of A_i, poses the most difficulties in implementation. Although we do not have a simple expression for the distribution of $A|F,s,\theta$, we can describe an algorithm for the generation of A. Fortunately, this is all that we need. Using the independence of the fragments, the assembly is generated on a row $\times$ row basis, with row i only dependent on fragment f_i. For a given f_i, we must generate an alignment $\vec{\alpha}_i$, a row vector of n_{α_i} elements, with each element taking values in $\{0,1,2\}$, as described by Churchill in Section 3.1.2. Taken together, we get a row vector $\{f_i,\vec{\alpha}_i\}$, of length n_{α_i}, where the component $\vec{\alpha}_i$ describes the alignment of fragment f_i with the clone sequence s (as shown in (4)). Note that each vector $\{f_i,\vec{\alpha}_i\}$ may have different lengths. The m vectors $\{f_i,\vec{\alpha}_i\}$. $i = 1,\cdots,m$ are then aligned together to form $\{A,\Gamma\}$ (see (5)), where gaps are inserted in each $\{f_i,\vec{\alpha}_i\}$ to correspond to gaps in s generated from $\{f_j,\vec{\alpha}_j\}$, $j \neq i$. Finally, the assembled fragments matrix X is obtained by deleting from $\{A,\Gamma\}$ all columns that contain only gaps. Therefore, the generation of X, the desired variable, follows directly from generation of each vector $\vec{\alpha}_i$. To generate $\vec{\alpha}_i$, we use the algorithm described in detail by Churchill in Sections 3.3.2 and 3.3.3.

4 Inference About the Clone Sequence

Once a sample of clone sequences $s^{(1)}, \cdots, s^{(k)}$ has been obtained from the Gibbs sampler, we can then combine this information into a composite "confidence" clone sequence. This would actually be a confidence set (or, more precisely, a Bayesian credible set) on the true sequence. Of course, it would be desirable to construct such a confidence set in an optimal manner, but it is not clear to us what the optimality criterion should be, or if an optimal construction is even feasible. We therefore content ourselves with presenting a method that leads to a usable confidence set, but almost certainly not an optimal set. We also point out some strategies for optimization. First, from the sample $s^{(1)}, \cdots, s^{(k)}$, identify the sequence s^M (the "modal" sequence) with the highest posterior probability,

$$P(s^M|F) = \max_i P(s^{(i)}|F).$$

Next, for a chosen distance function d, calculate d_i = distance between $s^{(i)}$ and s^M. Assume, without loss of generality, that $d_1 \leq d_2 \leq \cdots \leq d_{k-1}$. Then find the smallest value of k^* such that, for a specified confidence value 1–α,

$$P\left(\{s^M, s^{(1)}, \ldots, s^{(k^*)}\}|F\right) \geq 1 - \alpha$$

and take $s^C = \{s^M, s^{(1)}, \ldots, s^{(k^*)}\}$ as a 1-α confidence set. The set s^C can be written as a clone sequence with some ambiguous characters. For example, we might have a sequence s^C of length 8 given by

$$s^C = A|*|*|G|C|C \text{ or } T|*|T|$$

for a 95% confidence set. It is hoped that the elements of s^C will be less ambiguous where fragment alignment is unequivocal, and more ambiguous near the ends of the clone, where fragment alignment is more problematic. The probability calculations required to implement the algorithm are all straightforward, and follow directly from the Gibbs sampler. For each sequence $s^{(i)}$ we have

$$\begin{aligned} P\left(s^{(i)}|F\right) &\approx \frac{1}{k}\sum_{j=1}^{k} P\left(s^{(i)}|F, A, \Gamma_j, \theta_j\right) \\ &= \frac{1}{k}\sum_{j=1}^{k} \frac{P\left(\{s^{(i)}, \Gamma_j\}|F, A_j, \theta_j\right)}{P\left(\Gamma_j|F, A_j, \theta_j\right)} \\ &= \frac{1}{k}\sum_{j=1}^{k} \frac{P\left(\{s^{(i)}, \Gamma_j\}|X_j, \theta_j\right)}{P\left(\Gamma_j|X_j, \theta_j\right)} \end{aligned}$$

where X_j is the j-th assembled fragments matrix. The probabilities can now be calculated from the formulas in Section 3.1. The distance measure d can take many forms, but an optimal form is not known. For example, it might be reasonable to define the distance between $s^{(i)}$ and $s^{(j)}$ to be the number of non-matching bases. (A minor complication is caused by the fact that the alignment

of $s^{(i)}$ and $s^{(j)}$ is not well defined. This can be accommodated by calculating a minimum or maximum distance.) Other choices for the distance function can be based on a probabilistic weighting (giving higher weight to bases with higher probability) or methods based on the scoring schemes of Karlin et al. (1990).

5 Practical Aspects of Implementing the Chain

Each of the three steps of the Markov chain results in a straightforward algorithm for generating random variables from a posterior distribution. Thus, the Markov chain can be run to its stable distribution, and a sample of clone sequences $s^{(1)}, \cdots, s^{(k)}$ from $\pi(s|\ \mathrm{F})$ can be drawn. However, moving from theoretical calculations to practical implementation can often be extremely difficult, especially when a problem is of the magnitude of this one. We now address some of these difficulties.

5.1 Assessing the Computation Time

The first difficulty is the massive amount of computing that is necessary to run the Markov chain. This computing roadblock is quite impressive at this time, making the above Monte Carlo Markov chain (MCMC) algorithm formally available but, in practice, impossible to implement.

To make this point clearer, consider a sequence of 10,000 bases and 1,000 extracted fragments of average length 100 bases, sizes that would occur in practice. For each fragment, the computation of the probability matrix of §3.3 is of an order of $3 \times 100 \times 1,000 = 3 \times 10^5$. The simulation of A requires computations of the order of 3×10^8 (unless a faster method for simulating the alignment vectors, α_i, is discovered). Compared with this impressive amount of computation, the simulations of $\{s, \Gamma\}$ and of θ take a negligible amount of computation time. This implies that a single step of the Gibbs sampler requires 3×10^8 units of basic CPU time. If 3×10^8 is quite a manageable amount of operations for today's computers, in particular in parallel setups with each fragment being run separately, we must also take into account the fact that the Gibbs sampler requires a very large number of iterations to be efficient in this particular setup. In fact, although the sequence S takes values on a finite state space, $\{A, C, G, T\}^{\mathcal{G}}$, the cardinality of the state space is properly appalling since it reaches $4^{\mathcal{G}} \simeq 10^{6,000}$ when $\mathcal{G} = 10,000$. Obviously, there is no need to visit each element of the state space in order to achieve convergence of the Gibbs sampler but the complexity of the structure may require a very long computing time for convergence to occur, as well as a very careful monitoring to avoid fake convergence on subsets of the state space as they often appear in complex discrete settings (see below).

The finiteness of the state space guarantees proper convergence of the Gibbs sampler and most MCMC methods. Under modification of the algorithm (11) into

$$\begin{aligned} \{s, \Gamma\}^{(j)}, A^{(j)} &\sim \{s, \Gamma\}, A | F, \theta^{(j-1)} \\ \theta^{(j)} &\sim \theta | F, \{s, \Gamma\}^{(j)}, A^{(j)} \end{aligned} \tag{11}$$

the finiteness actually ensures geometric convergence of the Markov chain of the $(\{s,\Gamma\},A)$'s (Tierney, 1991) and thus of the chain $(\theta^{(j)})$ (Robert, 1993; Diebolt and Robert, 1993, 1994). But this theoretical reassurance is worth very little in practice since it does not give any hint at the actual speed of the algorithm. In fact, similar setups with huge finite state spaces, like those of the Ising model examined in Gelman and Rubin (1992), have pointed out the difficulty of attaining stationarity, as well as the dependency on startup conditions. So the theoretical picture associated with the Gibbs sampler in this setting is quite clear: since we can simulate any value of the sequence s from any previous value, the Markov chain $s^{(j)}$ produced by the Gibbs sampler is irreducible and aperiodic, and therefore recurrent and ergodic. There exists a single stationary distribution, the true posterior distribution of s, and convergence to this distribution is geometric and even φ-mixing. However, the practical behavior of the Gibbs sampler in such setups is pretty much unknown. For one thing, the (conditional) probabilities of most states of the sequence must be negligible, but the width of the state space is such that the significant values cannot be identified easily and, more importantly, that it will usually require a considerable number of iterations to move from one mode of the posterior distribution to another, i.e., to explore thoroughly the posterior surface. It is then quite likely that "apparent stabilization" phenomena, as those exhibited in Gelman and Rubin (1992), can occur.

The assembly line inertia, that is, the propensity for the chain to remain in a nearby state, is also likely to be quite high under the Gibbs sampler perturbations. Consider the case of a single fragment f_i being misplaced. The number of iterations required for the actual position of f_i to occur can be quite high if the present location of f_i has an influential effect on the corresponding sequence, i.e., if the present position of f_i is then much more probable under the simulated s than its true position. This difficulty is obviously magnified by the number of possible starting values for a given fragment, roughly $(2\mathcal{G})^{|f|}$ if the sequence is of length $\mathcal{G}$, and, evidently, by the number of fragments. We are again facing huge numbers, of the order 10^{400}. It could be interesting to make use of the present deterministic techniques of sequencing to derive a starting point for the algorithm or, on the contrary, to see how many iterations of the Gibbs sampler are required to see this sequence (or a close modification) appear. Starting with existing techniques makes the Gibbs sampler appear as another type of *simulated annealing*, i.e. of a technique where deterministic solutions are randomly perturbed to see whether more interesting solutions exist in their neighborhood.

5.2 Alternative Algorithms

As noted above, more advanced convergence results can be established for the modification (11) of the algorithm. The direct simulation of $\{s,\Gamma\},A$ can be based on the marginal

$$\pi(\{s,\Gamma\}|F,\theta) \propto \frac{\pi(\{s,\Gamma\}|F,A,\theta)}{\pi(A|F,\theta,\{s,\Gamma\})} \tag{12}$$

and on the conditional distribution $\pi(A|F,\theta,\{s,\Gamma\})$. Note that, although

the ratio involves A, the density (12) is independent of A.

However, a more general computing alternative to the algorithm proposed by Churchill is to call for another MCMC device. Although Gibbs sampling is often the most natural and the simplest choice of MCMC algorithm in a Bayesian setup –and it certainly is in this case–, Gibbs sampling can suffer from slow convergence properties in complex settings due to difficulty in escaping local modes of the posterior distribution. More "energetic" perturbations, like those induced by the *Metropolis-Hasting* algorithm (see Tierney, 1991), may be more appropriate. In fact, while exploring more thoroughly the parameter space, this algorithm simultaneously reduces the computing time required for each iteration. The distribution from which the fragments are simulated (the "working" distribution) may be selected for both ease of simulation and analytical tractability. In such cases the computation time can be cut down considerably. For example, for a fragment f, while the simulation time is of order $|f|$, computing the probability weight involved in the Metropolis acceptance step is also of that order since we only need to follow the path corresponding to f in the matrix of Figure 4. The Metropolis scheme can also be implemented in the case of the modification (12) since $\{s, \Gamma\}$ can be simulated according to a manageable distribution based on the previous value $\{s, \Gamma\}^{(j-1)}$, $g(\{s, \Gamma\}|\{s, \Gamma\}^{(j-1)})$, and the simulated value accepted as $\{s, \Gamma\}^{(j)}$ with probability

$$\varrho = \frac{\pi(\{s, \Gamma\}|F, A^{(j)}, \theta)\pi(A^{(j)}|F, \theta, \{s, \Gamma\}^{(j-1)})\, g(\{s, \Gamma\}^{(j-1)}|\{s, \Gamma\})}{\pi(\{s, \Gamma\}^{(j-1)}|F, A^{(j)}, \theta)\pi(A^{(j)}|F, \theta, \{s, \Gamma\})\, g(\{s, \Gamma\}|\{s, \Gamma\}^{(j-1)})} \wedge 1.$$

Again, note that the ratio ϱ is actually independent of $A^{(j)}$ despite the notation.

Along with this saving by a factor 10^3 and the apparent simplicity of the Metropolis algorithm, there are also disadvantages to the Metropolis approach. Although replacement of a "true" simulation step in a Gibbs algorithm by a Metropolis approximation retains the same convergence properties as the original chain, the chain generated by the Metropolis algorithm can remain at a certain spot for very long time if the Metropolis weights are too small to induce a change. However, this criticism applies to every implementation of this algorithm.

In practice, the Metropolis algorithm is often less likely to get stuck than a regular Gibbs sampler because of the larger scale of random perturbations it involves. For Churchill's setting of DNA sequences, we can even suggest some approaches to the implementation of the Metropolis approximation. In fact, the working distribution can be chosen as a random walk perturbation of the previous alignment of each fragment. The parameters of this random walk have to be chosen with care since, if they induce too small a variation in the chain, potential trapping states may appear for the simulated chain (although they are impossible theoretically). Alternatively, if the corresponding variance is too large, the chain will be too perturbed to achieve any visible stationarity. We therefore suggest a preliminary tuning of these parameters in the first (1,000? 10,000?) iterations of the algorithm in order to achieve a range of rejection between 30% and 70% as in Muller (1992) or Besag and Mengersen (1993). Once

this stable mode of perturbation is reached, the Metropolis algorithm can then be modified into an hybrid Metropolis algorithm, with two different magnitudes of perturbation. In fact, as the algorithm approaches the mode of the posterior distribution, large variations between simulations actually slow convergence. Therefore, we suggest the use of occasional shake-ups in the simulation, with intermediate moderate perturbations, in order to preserve global modes but also to ensure a 'uniform' coverage of the parameter space.

5.3 The Number of Paths

A last, somewhat more technical, remark. Gary Churchill inquired during his talk about the number of paths through a p by n matrix when the only possible moves from (i,j) are to $(i,j+1)$, $(i+1,j)$ and $(i+1,j+1)$. This number is actually

$$\sum_{k=0}^{n\wedge p} \binom{n+p-k}{k} \binom{n+p-2k}{p-k} = \sum_{k=0}^{n\wedge p} \frac{(n+p-k)!}{k!(p-k)!(n-k)!}, \tag{13}$$

where $n \wedge p$ is the minimum of n and p. To see that (13) actually holds, consider that a walk in the matrix according to the above rule is a sequence of 'r' (for right), 'a' (for across) and 'd' (for down), depending on whether it goes from (i,j) to $(i,j+1)$, $(i+1,j)$ or $(i+1,j+1)$. If R denotes the number of 'r', D the number of 'd' and A the number of 'a', the constraints on A, D and R are

$$0 \le A \le n \wedge p, \quad 0 \le D \le p, \quad 0 \le R \le n,$$

while $A + D = p$ and $R + A = n$. This implies that $0 \le A \le n \wedge p$ and that R and D are determined by A. For a given k, if $A = k$, the number of terms in the sequence is then $k + (p-k) + (n-k) = p + n - k$ and the number of different allocations of the 'A' steps is $\binom{k}{p+n-k}$ and, in the remaining $(p+n-k) - k = p + n - 2k$ spots, the number of different allocations of the 'R' is $\binom{p+n-2k}{n-k}$. The total number of paths is then indeed

$$\frac{(n+p-k)!}{k!(p-k)!(n-k)!}$$

for a given k.

6 Conclusions

As a coincidence, a paper appeared in *Science* the very week of the conference, by Lawrence et al., which is very closely related to Churchill's paper, and lays the first steps of an actual implementation of Gibbs in DNA recognition. We want to mention the results contained in this paper since (a) it has been published in a Biology journal, not commonly read by statisticians, and (b) it somewhat moderates the above conclusion about the practicality of Churchill's results.

Lawrence et al. (1993) deal with protein, rather than DNA, recognition, with the main difference being that the state space for a single point of the sequence is now of size 20 instead of being restricted to 4 points. In contrast to Churchill's analysis, Lawrence et al. are concerned with multiple alignment (alignment of several sequences) between different species, rather than reconstruction of a single sequence from fragments. The reason for this study is to exhibit a common protein structure, as long as possible, and allowing for a certain amount of discrepancy between different species. This search for common patterns is supported by an evolutionary theory of common ancestry (which we cannot describe here). The important point is that the different sequences to be compared are assumed to be already perfectly known. In the alignment to be realized, the differences are thus explained by evolution and specificity of each species, not by chance errors in reading.

Using first a fixed size N for the almost common sequence of proteins, Lawrence et al. propose a 'Gibbs-like' algorithm which is linear in the number of sequences and in N. Their algorithm is actually closer to a stochastic perturbation of the EM algorithm, the SEM algorithm (Celeux and Diebolt, 1986; Wei and Tanner, 1990; Qian and Titterington, 1991; Robert, 1992), than to exact Gibbs sampling. To be more precise, the authors do not simulate from the conditional posterior distribution of the parameters but rather take the corresponding expectations as current values for the next simulations of the alignment. Their generation of an alignment is not exact either, as they replace the conditional posterior probability of a given alignment by its odds ratio. But their approach could be directly translated into a true Gibbs algorithm, with the addition of a distribution on the length of the common sequence.

However, in contrast to Churchill's algorithm, this algorithm works linearly because the authors do not allow for gaps or insertions, but only for differences in the protein sequences. The matrix of §3.3 can then be read linearly and forward so the computation times become quite reasonable. For example, for 20 sequences of length varying from 20 to 512, convergence was attained in 1000 to 3000 iterations, (except in some cases where a suboptimal trapping state was reached).

It is thus very exciting to see variants of Gibbs sampling already implemented in practice for DNA identification. Among other things, this shows an increasing awareness of the need for more elaborate alignment methods. In addition, it may be possible to use such an approach in the warm-up steps of Churchill's algorithm, by deriving an alignment where insertions and deletions would be first omitted. More accurate MCMC algorithms could thus start from this crude alignment, which could provide a better starting point and hence lead to convergence in a reasonable time.

Finally, our overall conclusion is extremely positive. The model proposed by Churchill results in an implementable Markov chain whose output can be used to provide a valid inference about the clone sequence, including an assessment of confidence. Moreover, there is great flexibility in the underlying parameter structure which allows the model to better reflect the real process. The only drawback, if it can even be considered so, is complexity. Using data of realistic size, there are too many calculations necessary to expect usable output in our

lifetimes. However, this "drawback" is a red herring for two reasons. One, with computing speed increasing every day, this algorithm soon may be computable. But second, and more important, this algorithm and model represents a "gold standard". We now need to develop approximations and faster random variable generations that can be tested against the gold standard in small data sets, and are computationally feasible in realistic data sets.

Additional References

Besag, J. and Mengersen, K.L. (1993) Meta-Analysis using Monte Carlo Markov Chain methods. Tech. report, Dept. of Statistics, Colorado State Univ.

Celeux, G. and Diebolt, J. (1986) The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Comput. Statist. Quater.* **2**, 73-82.

Diebolt, J. and Robert, C.P. (1993) The Duality Principle: Discussion of Smith and Roberts, Besag and Green, and Gilks *et al.*. *J.R.S.S.* (Ser. B) **55**, 73-74.

Diebolt, J. and Robert, C.P. (1994) Estimation of finite mixture distributions by Bayesian sampling. *J.R.S.S.* (Ser. B) **56**, 163-175.

Gelman, A. and Rubin, D.B. (1992) Does a single iteration suffice? In *Bayesian Statistics 4* (J.O. Berger, J.M. Bernardo, A.P. Dawid and A.F.M. Smith, eds.) Oxford University Press, London.

Karlin, S., Dembo, A., and Kawabata, T. (1990). Statistical composition of high-scoring segments from molecular sequences. *Ann. Statist. 18*, 571-581.

Lawrence, C.E., Atschul, S.F., Boguski, M.S., Liu, J.S., Neuwald. A.F. and Wootton, J.C. (1993) Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science* **262**, 208-214.

Muller, P. (1992) A black-box algorithm for implementing the Metropolis algorithm. Tech. Report, Dept. of Statistics, Purdue University, Lafayette.

Qian, W. and Titterington, D.M. (1991) Estimation of parameters in hidden Markov models. *Phil. Trans. Roy. Soc. London* A **337**, 407-428.

Robert, C.P. (1992) Discussion of Meng and Rubin In *Bayesian Statistics 4* (J.O. Berger, J.M. Bernardo, A.P. Dawid and A.F.M. Smith, eds.) Oxford University Press, London.

Robert, C.P. (1993) Convergence assessments for Monte-Carlo Markov chain methods. Technical Report, Dept. of Math, Univ. de Rouen.

Tierney, L. (1991) Markov chains for exploring posterior distributions. *Computer Sciences and Statistics: Proc. 23d Symp. Interface*, 563-570.

DISCUSSION

Kathryn Roeder. Carnegie Mellon University

1 Introduction

I would like to congratulate Professor Churchill for presenting an excellent introduction to the physical mapping problem and providing a promising Bayesian model for DNA sequence alignment. Research on this problem, which is inherently statistical, has been dominated by computer scientists and mathematicians. Although these researchers have developed useful algorithms for sequence reconstruction, they have largely ignored the issue of errors in alignment. Most molecular biologists recognize that errors are present in the alignments, but they have never quantified the amount. None of the alignment techniques currently in use indicate the uncertainty of alignment, by position. Furthermore, equally plausible alignments are seldom examined. Thus the problem is ripe for statistical treatment. Nevertheless, a careful study of the proposed Bayesian method will reveal that the problem is too complex to be solved without major modification of the usual Bayesian Markov sampling techniques. Clearly this is a challenging and important problem that deserves further attention from the statistical community.

During the conference, a number of persons asked a vexing question: "Why do we want to read the DNA sequence anyway?" In fact, this is a question asked by some well-known geneticists also. A vocal minority think that this project, like many other big science projects, is a misguided effort. They argue that knowing the DNA code will not add to our understanding and treatment of genetic diseases or answer other basic genetic questions. This argument, however, is being challenged almost daily by creative geneticists who use this kind of detailed information to treat human diseases, answer riddles of evolution, and the like.

Other scientists object to physically mapping the genome on the basis that there are smarter ways to read the genetic code than the "Humptydumpty" method explained in this article. The competing school of thought supports a targeted approach known as genetic mapping (as opposed to physical mapping). With genetic mapping, key areas of the genome are identified as likely to be associated with certain functions, for example, coding for an enzyme. Given this alleged association, a targeted effort is made to decode this particular stretch of DNA. This is the school of thought to which I personally subscribe.

Nevertheless, reading the complete genome is the final goal of the human genome initiative. Moreover, regardless of the approach taken, eventually one

must reconstruct the sequence in relatively small blocks of the genome. Once a region of the genome is targeted as being the likely location of a disease gene, it is particularly important that the sequence be reconstructed with no errors because some diseases are characterized by a single error in the sequence. Given that the problem of sequence alignment has a wide level of intellectual (and monetary) support, it behooves the statistical community to contribute to the effort where possible.

2 Errors in the fragments

The molecular technique by which a sequence is read, which was pioneered by Sanger et. al. (1977), is described by Churchill in Section 1.2.2. Briefly, this technique involves cutting up identical copies of the DNA sequence at specific positions in the sequence, each cut site being the location of the same nucleotide (or one of the four bases). Because the cut sites occur after the specified nucleotide, larger DNA fragments result for later positions in the sequence. The fragments are subsequently separated and the bases "read" by gel electrophoresis. Large molecules travel short distances and small molecules travel longer distances on an electrophoretic gel. Those of us lucky enough to see an actual gel at the conference can easily imagine that those bases on the "bottom" of the gel will be read with greater error than those on the "top" of the gel. The reason for this differential error rate is simple to understand though. Large molecules do not separate out on the gel as well as shorter molecules, and hence there is a great deal of error involved in reading the bases associated with these large molecules.

My objective is to quantify the error rate by position. This discussion deals only with Churchill's I and R states. The process is in the I state when an extra base is read; the observation is called an insertion. The process is in the R state when a base that exists in the true sequence is read; if this base is read incorrectly, this is called a misread. Given that the process is producing an output, errors can be classified into two basic categories: insertions and misreads. Misreads can be further classified as mismatches (read a instead of b), ambiguous (unreadable base) and deletions (a gap in the output). It is natural and correct to infer that there is some ambiguity between deletions and insertions. When only two fragments are available from which a consensus sequence is to be inferred, it is impossible to ascertain whether one sequence has an insertion or the other has a deletion. Generally, more than two sequences are available from which a consensus can be determined; hence it is possible to make a good guess as to whether the gap is due to an insertion in one or more fragments or a deletion in one or more fragments.

The data made available to me consist of a compilation of errors by position inferred from approximately 20 assemblies. An assembly was constructed from a set of fragments that are assumed to be measurements of the same stretch of DNA (see Churchill's Figure 5). These fragments were aligned using a standard sequence alignment algorithm (see Waterman [1984] for an overview of available algorithms). First, regions of high similarity are selected using a hashing

method. Next, pairs of fragments are aligned using a dynamic programming algorithm that gives a positive score for matching bases, and a negative score for mismatches and gaps. This algorithm yields an alignment between a pair of fragments with order $n \log n$ calculations. The result is not necessarily even the alignment that maximizes the score; however, because the alignment can be calculated in real time, it is popular. To align multiple fragments, the same procedure is repeated. The next sequence is aligned to the consensus of the previous pair of sequences and so on with some modifications to remove the effect of the order of the procedure.

Once the fragments are aligned, an overall consensus is determined based on majority rule. For example, in the first position of the fifth block of data presented in Churchill's Figure 5, the consensus is clearly A since we observed $AAAAT$ in that column. Further along that block a deletion appears (the column reads $AA-AA$) and then four positions later an insertion occurs ($--A---$). When the aligned fragments are only one deep, no errors can be detected. When the aligned fragments are only two deep, a gap in one of the two fragments is somewhat arbitrarily classified as a deletion rather than an insertion. In the portions of the assembly for which the aligned fragments are 6 or more deep, the consensus declaration does not depend on many judgment calls. Although there is little doubt that some portions of the assemblies are incorrect, I will treat the consensus obtained in this way as the true sequence from which the error rate, by position, can be estimated

Table 1 illustrates the combined information on error rates at position 600. This information is assimilated from the available assemblies. The data are organized by the position on the fragment as it is read, not the position of the consensus. Each of the fragments that yield output at position 600 is compared to its respective consensus sequence to determine if an error occurred. At position 600, 188 fragments were observed of which 182 corresponded to proper base declarations ($\{A, C, G, T\}$, 1st 4 rows of Table 1). Of these 182 observations, 3 were mismatches (off-diagonals in 1st 4 columns), 3 were ambiguous (5th column, first 4 rows), and 17 were deletions (6th column, first 4 rows). The overall probability of a misread is then estimated as 23/182 at position 600. An insertion has occurred if the consensus is '-' when the fragment registers a base of some kind ($\{A, C, G, T, N\}$). At position 600, the probability of insertion is estimated at 3/168 (6th row, first 5 columns).

	fragment						
consensus	A	C	G	T	N	-	total
A	45	0	1	0	0	4	50
C	0	34	0	0	2	3	39
G	0	0	35	1	1	5	42
T	0	1	0	45	0	5	51
N	0	0	0	0	0	0	0
-	0	2	0	0	1	3	6
total	45	37	36	46	4	20	135

Table 1: Cross tabulation of observed bases in fragments, by consensus, at position 600. Data were obtained from the compilation of approximately 20 assemblies, obtained from Churchill.

If the error rate depends smoothly on position, one can use a nonparametric regression technique to estimate error rate by position. For these data, a natural choice is Loess (Cleveland 1979). Let Y_k and n_k denote the number of errors and the sample size at position k. A local quadratic regression of $\hat{p}_k = \frac{Y_k}{n_k}$ on k, using a span of 1/4 and weight equal to n_k, provides a good fit to the data. Figure 1a illustrates the mismatch and insertion probability (times 100) estimated in this way. Just as postulated, the error rates are quite low for positions early in the sequence (short fragments), but increase dramatically by position. Figure 1b illustrates another interesting feature of the data. Very few of the fragments are long enough to be extremely error prone.

Figure 2 compares the four types of errors by position. Notice that although insertions increase earlier in the sequence ($k \approx 375$), it is deletions that dominate, rising up to a level of 25%. This analysis corresponds well with a similar analysis of assembly data conducted by Koop et al. (1993), except that they obtained a reversal of deletions and insertions. Presumably the curves were mislabeled in their analysis.

The standard error bands of each estimate are depicted in Figure 3. As one would expect from Figure 1b, the variance increases dramatically with position. Clearly little information on error rates is available past position 700. However, the error rate must continue to increase beyond that position, and therefore there will be little worthwhile information obtained from long fragments. One should not be misled by the Loess estimates, which eventually decrease as k increases past position 700. Not only is this bias characteristic of Loess estimates, but it is inherent in the assemblies as well, because some errors are not detected in the sparse portions of the assembly.

To determine if the curves were over-smoothed, residuals were plotted. Let p_k^ℓ denote the Loess estimate at position k and let $v_k = p_k^\ell(1 - p_k^\ell)/n_k$ denote the estimated binomial variance of $\hat{p}_k$. The residual at position k is defined as $z_k = (\hat{p}_k - p_k^\ell)/\sqrt{v_k}$. If the Loess estimator is not over-smoothed, then z_k should be approximately distributed as a standard normal. Because the residuals depicted in Figure 4 show no deviations from the model, one can conclude that the choice of smoothing parameter was reasonable.

Curiously, specific errors are more likely than others. From Figure 5, an interesting pattern emerges. Once insertions become common, C and G are inserted with greater probability than T, A and N. Moreover this insertion rate is nearly equal for the two groups ($\{C, G\}$ vs. $\{A, T, N\}$.) The same pattern emerges for mismatches (Figure 6). Of the twelve types of specific mismatches ($Pr(\text{observe } a|\text{consensus is } b), a \neq b$), four types are more common: $\{C|A, G|A, C|T, G|T\}$. Again the specific error rates split into two groups with nearly equal behavior within a group. The reason for the preponderance of Cs and Gs is not clear, but there is likely to be a biological explanation for this regular pattern of errors.

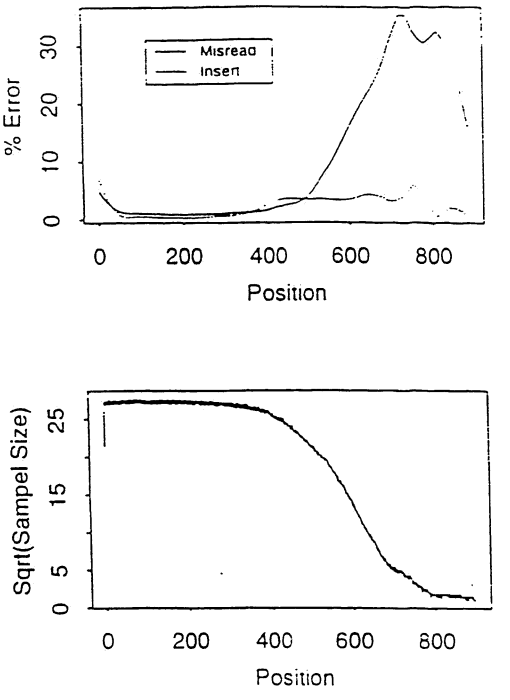

Figure 1. Percent Error and sample size, by position. The top figure (a) illustrates percent insertion and percent misread (mismatch + deletion + ambiguous) as a function of position of the fragment. The bottom figure (b) illustrates square root of sample size as a position of position.

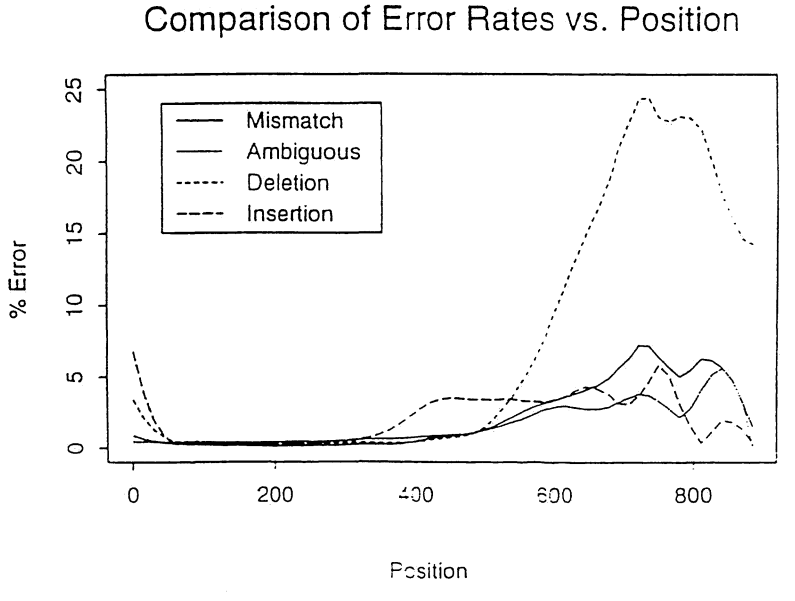

Figure 2. Percent error (insertion, deletion, mismatch and ambiguous) by position.

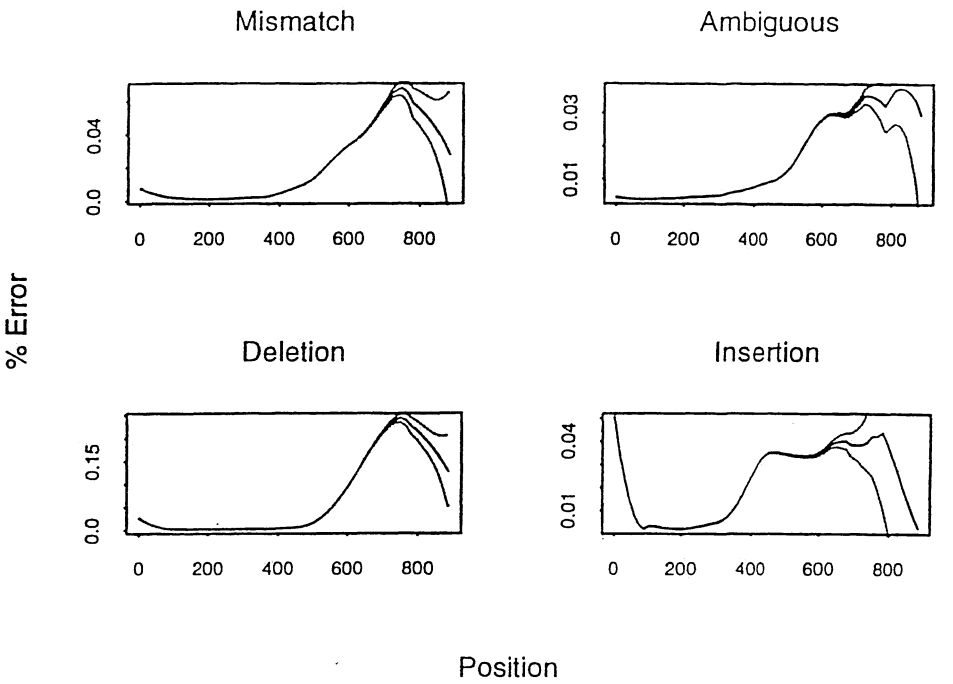

Figure 3. Percent error (insertion, deletion, mismatch and ambiguous) by position, with error bars placed at plus or minus three standard deviations.

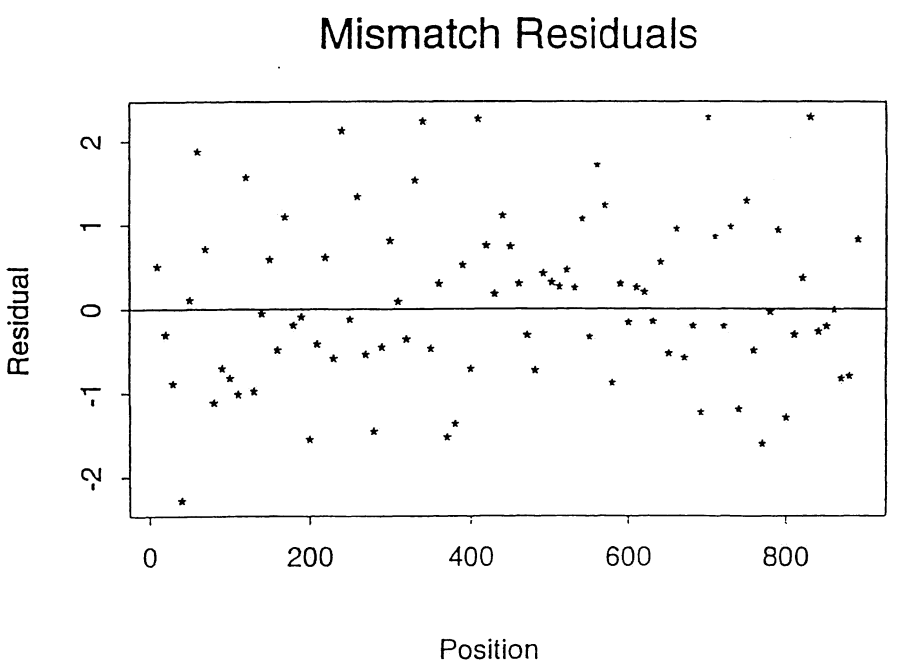

Figure 4. Standardized residuals for the mismatch probability estimates.

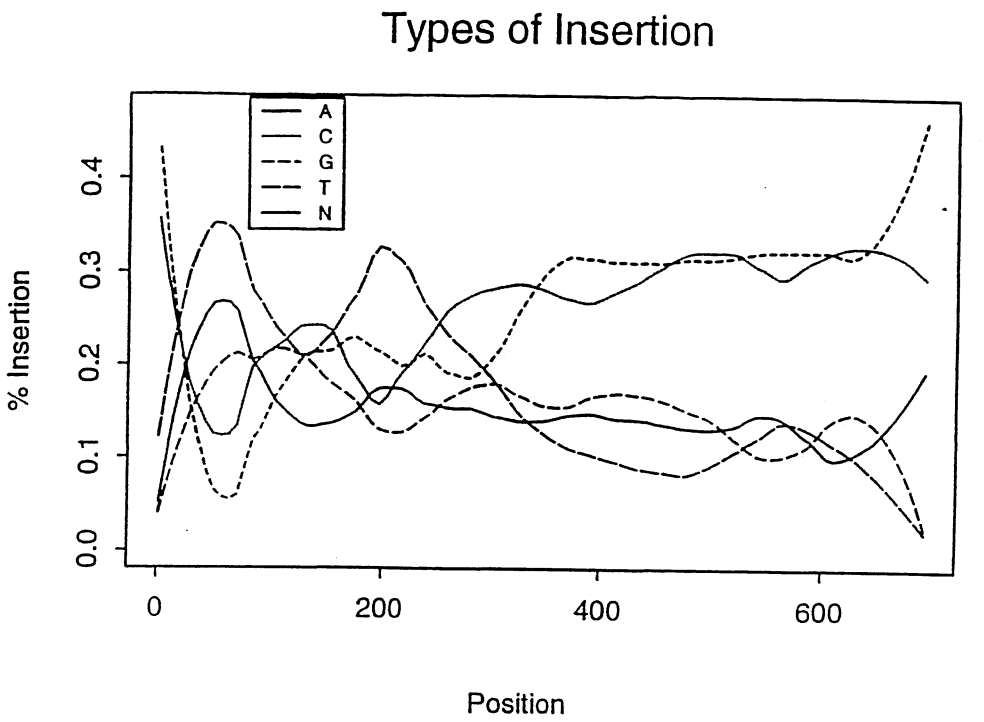

Figure 5. Percent insertion of each type (A,T,C,G,N), by position.

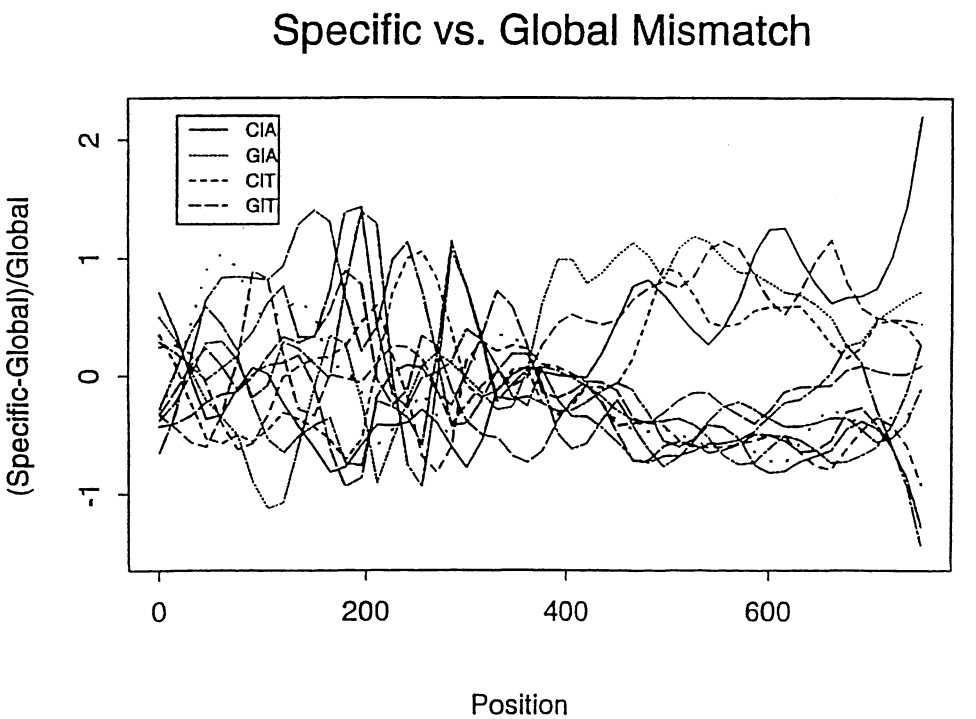

Figure 6. Relative mismatch probabilities. The specific error probability is $Pr(\text{read}a|\text{concensus}b)$, $a \neq b$. The global mismatch probability is obtained by dividing the nonspecific mismatch probability by three. The relative error rate is the difference between the specific and the global error rate, divided by the global error rate.

3 Conclusions

From this analysis of error rates, it is apparent that any method for sequence alignment that ignores the fragment position is flawed. Error rates towards the end of a sequence are surprisingly large compared to the small error rates early on in the sequence. Notably, the deletion rate ranges from less than 1% for $75 < k < 475$ to over 20% for $k > 650$. Methods that ignore differential error rates could be used only when truncated (small) fragments are used.

A Dirichlet prior could be constructed from these data. For instance, the error probabilities could be estimated using Loess, with the scale parameter equal to the size of the training data set, by position. Such a prior would be highly informative. Information concerning rates of change from one state to another, available in these assemblies, could also be used.

Estimates of prior errors will be affected by the quality of the assemblies, but presumably consistent estimates of error rates could be obtained by iteratively realigning these data using the Churchill method. It is possible that the high error rate estimated for larger fragments is due to misalignment of the data in the tails of the sequences. This hypothesis could be tested with a larger database.

Of course, incorporating a prior is still hypothetical since the methods outlined by Churchill are not feasible yet. His methods are prohibitively expensive in terms of computations. Even though a perfect implementation of the Gibbs sampler is impossible, Churchill and colleagues may find an ad-hoc implementation profitable. Geneticists are already proceeding with the imperfect assemblies obtained using dynamic programming algorithms. Any effort that quantifies the uncertainity of the alignments would be a significant contribution. A less ambitious effort might involve aligning only the high quality portion of the fragments for which errors are unlikely and the probability of errors and the rates of change between states is constant.

The biggest obstacle to a successful implementation of Churchill's model is the multimodality of the likelihood surface. Because the parameter space is extremely large, this problem is even more daunting than usual. A fragment that is mislocated will be extremely hard to move to a radically different portion of the genome using Gibbs sampling. It is well known that the Gibbs sampler tends to move slowly, if at all, toward a new mode. Unfortunately one would expect such major misplacement of DNA sequence fragments. Frequently the same pattern of DNA appears in several portions of the genome.

Before Professor Churchill's efforts, the methods available to determine DNA sequences had progressed very little beyond the dynamic programming paradigm. His work on the hidden Markov model takes the field of DNA sequencing in a radically new direction: one that shuld be immensely valuable. I congratulate him for his efforts.

References

Cleveland, W.S. (1979) Robust Locally-weighted Regression and Smoothing Scatterplots. *J. Amer. Statist. Assoc.* 74, 829-836.

Koop, B.F., Rowan, L., Chen, W.-Q., Deshpande, P., Lee, H. and Hood, L. (1993). Sequence Length and Error Analysis of Sequenase and Automated *Taq* Cycle Sequencing Methods. *Biotechniques* 14,442-447.

Sanger, F., Nicklen, S. and Coulson, A.R. (1977). DNA Sequencing with Chain Terminating Inhibiters. *Biochemistry* 74, 560-564.

Waterman, M.S. (1984). General Methods of Sequence Comparison. *Bull. Math. Biol.* 46, 473-500.

Reply to the Discussions of Roeder and Casella/Robert

First I would like to thank the discussants for their insightful comments and for the effort they have put into deciphering the details of the algorithm I have described. I would like to address some of the implications of their comments especially with regard to the practical implementation of this approach to the estimation of sequences and sequence accuracy. Finally I would like to impress upon the reader that despite the discouraging complexity of the sequencing problem, there are a number of interesting pieces of the puzzle that remain unsolved and whose solution would be of both practical and theoretical importance.

I thank Drs. Casella and Robert for providing a clear and succinct summary of the Gibbs algorithm and for pointing out important details that were overlooked in my description of the algorithm. A new and important issue raised by Casella and Robert is the question of "what is to be done with the output of the Gibbs chain?". They propose a method of constructing confidence sets by defining a neighborhood around a modal sequence. In general the problem of constructing sets on sequence spaces is an open problem that should be pursued further. The approach described is promising but further investigation into appropriate metrics and methods of constructing sets is needed.

With regard to the number of paths through an alignment path graph: I thank Casella and Robert for the clarification but I feel that my response to the question is correct as it stands. There are lots of paths.

With regard to the practicality of the algorithm, I offer no excuses. My intention was to develop a rigorous solution to the problem with only a vague hope that it might prove to be practical. Although the prospect of 10^8 calculations per iteration is not too daunting with modern computing, the size of the the state space and relative rigidity of the simple Gibbs sampler are discouraging. I remain optimistic that approximate methods can be developed to replace the alignment stage of the sampler. This optimism is based on the observation that in real assemblies of fragments, there are many large regions that are unambiguous and relatively error-free. The regions containing errors and ambiguities can thus be isolated and the computational burden greatly reduced by sampling the alignments locally. It is also often observed that there are whole

fragments that are troublesome, either due to misalignment or some failure of the particular sequencing reaction. A combination of approaches that sample alignments locally, e.g. within a sliding window, with a process that shifts or removes whole fragments might be feasible. The success of a local alignment strategy will require that the gross structure of the assembly is correct. It may be possible to confirm this by other experimental methods.

The discussion by Dr. Roeder also makes note of the computational complexity of the alignment problem in its full implementation. However, the important contribution of her discussion is in the critical examination of the goodness of fit of the model to the observed data. In particular Dr. Roeder has looked at the distribution of errors (and different error types) as a function of the position within a fragment. The failure to account for position-dependent error rates is perhaps the biggest shortcoming of the model I have proposed. The uniformity of errors is clearly violated and it seems intuitively that this lack of uniformity could bias estimates of sequence accuracy. Position-dependent variation of error rates is a well recognized phenomenon. The key contribution here is the idea that error rates should vary smoothly as a function of position. It may be possible to develop hierarchical models which relax the uniformity assumption without going to the extreme of allowing independent error rates at each position. The importance of looking at the data should be emphasized here. Empirical knowledge of the relative rates of different error types and their dependence on position within a fragment are potentially useful for the resolution of ambiguities in the sequence. For example, we could know which of the bases is most likely to be in error when an ambiguity occurs in a two–deep region of an assembly. As new sequencing strategies are developed, it is likely that low–coverage directed methods will become more common, thus increasing the importance of prior knowledge of error types and locations in the estimation of sequences.

As a final comment, I want to point out that hidden Markov models provide a rich class of probablity distributions on sequence spaces. Statistical inference methods for HMMs are relatively underdeveloped and the potential for applications extends well beyond the sequencing problems described here. I would like to note that the Gibbs algorithm of Lawrence et al. (1993) is in fact (implicitly) operating on a HMM. The work of Krogh et al. (1994) has already been cited as another important application of HMMs to sequence data. The full potential of HMMs to address problems and questions related to DNA and protein sequence data has yet to be determined. I expect that over the next several years applications of HMMs will develop rapidly and some general principles of inference for HMMs will begin to emerge.